commodore home computing power play

Volume 3, Number 2, Issue 9
June/July, 1984
$2.50 U.S. $3.50 Canada
ISSN 0739-8018

The Best Chess for the 64

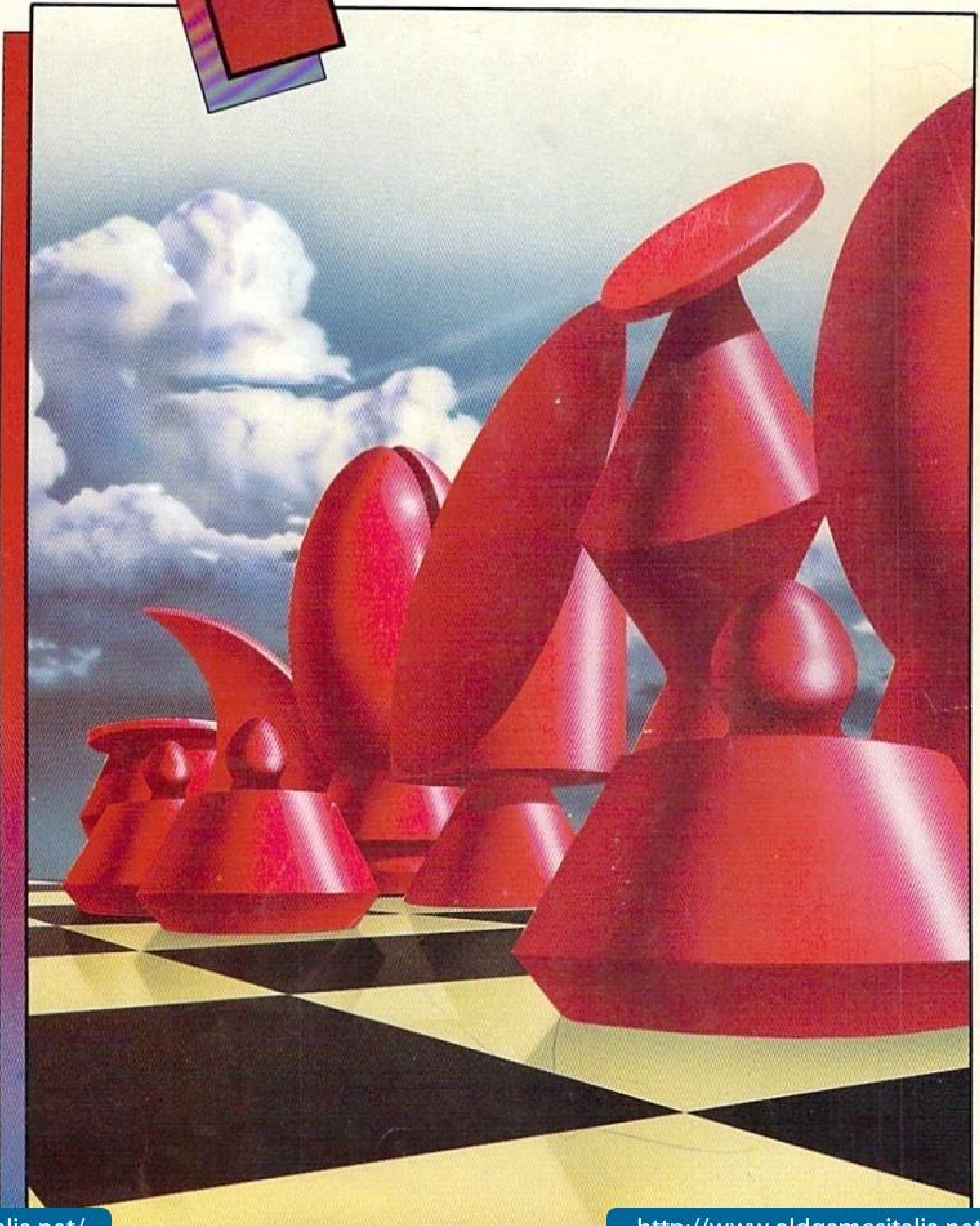
Can You Survive M.U.L.E.?

For Bridge Players Only

Suicide Mission on the Russian Front: Panzer-JAGD

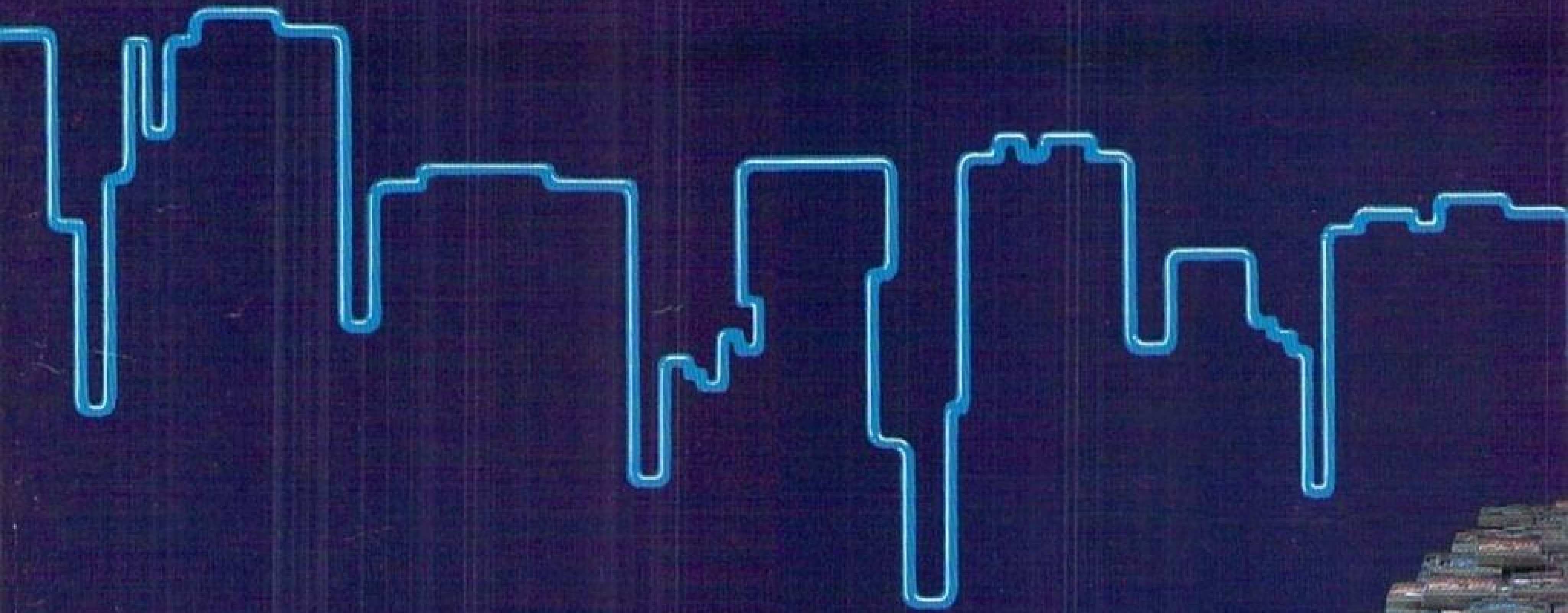
Telengard: An Adventure in Terror

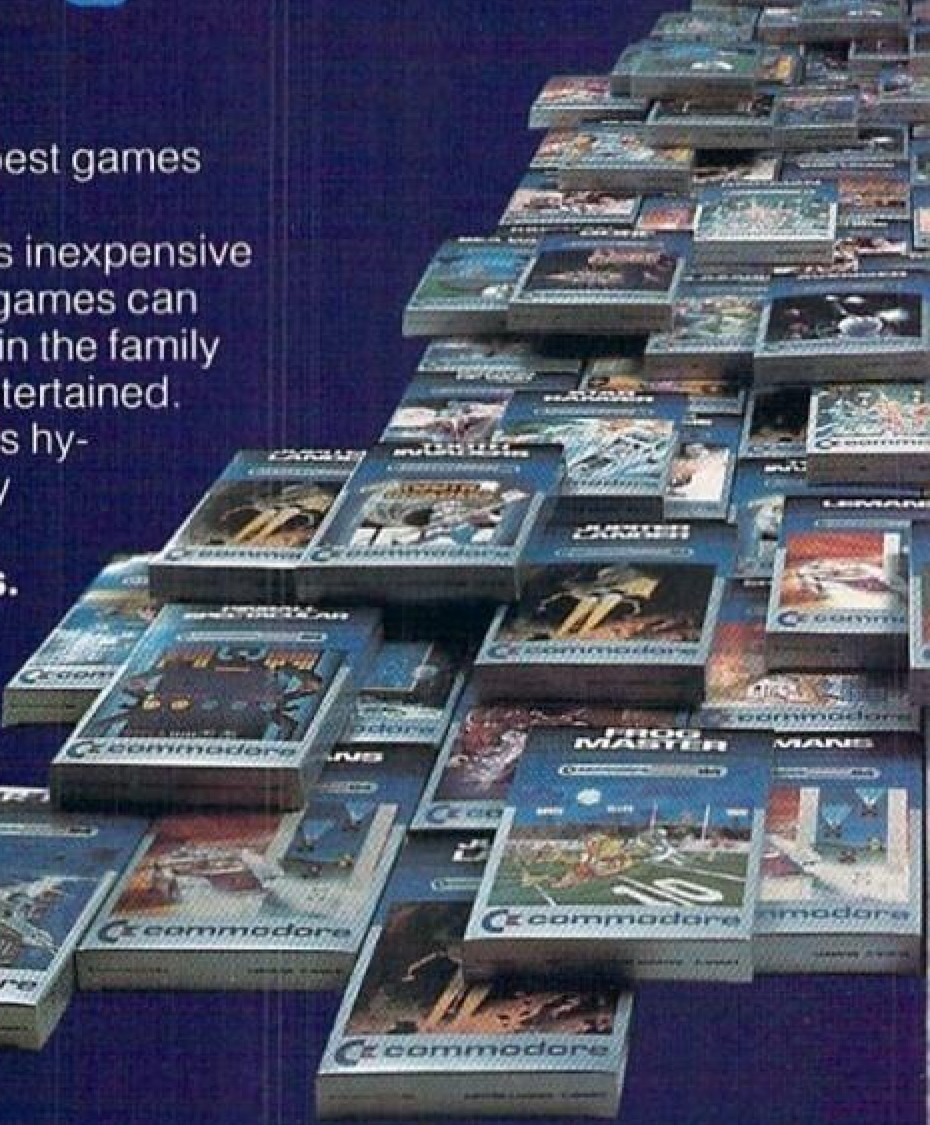
Archon: Chess, Magic and Monsters



Commodore Software–

...Take on the world, toughen up your trigger finger and fire away...





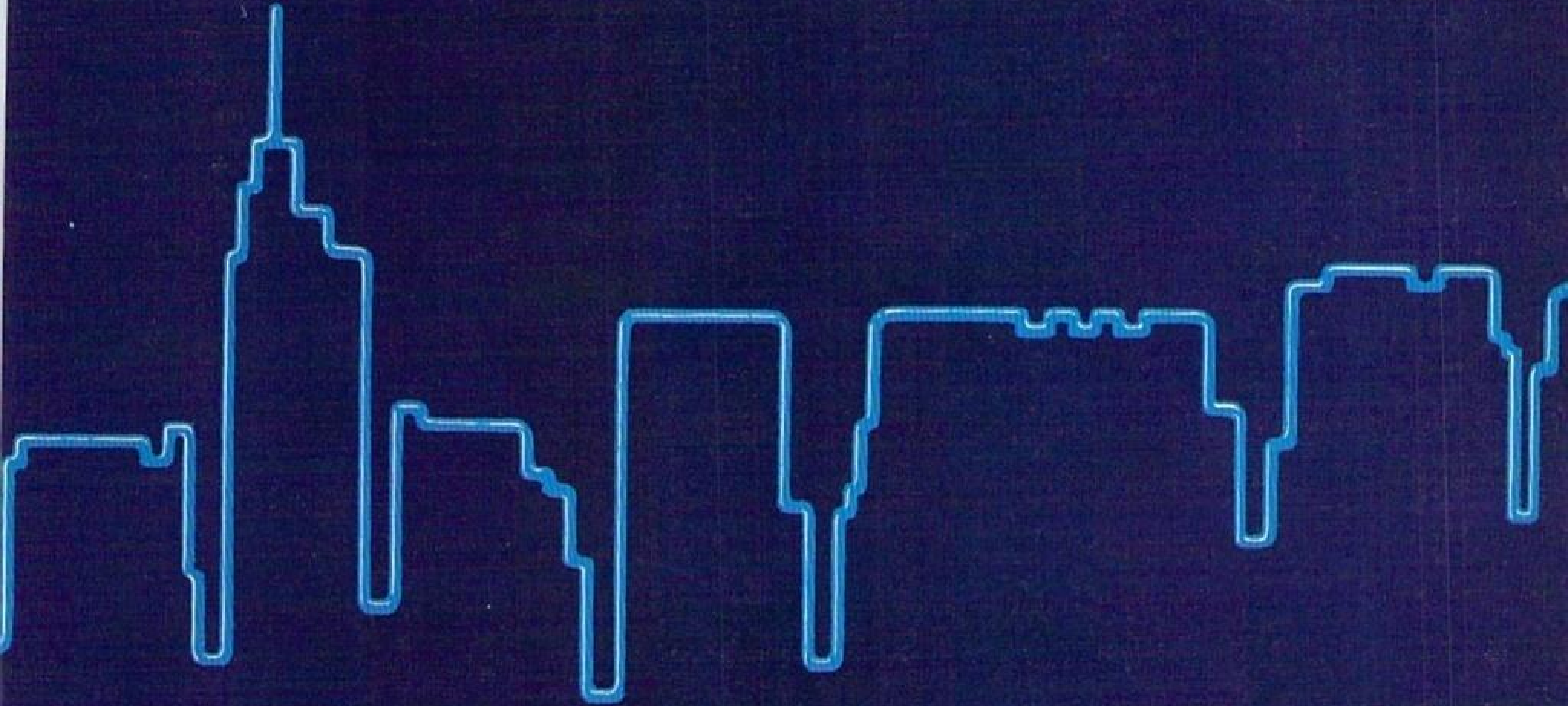
Commodore has been the best game in town in computers for awhile... at home, at school and work... with our exciting, easy to use, inexpensive VIC 20 and C64 computers.

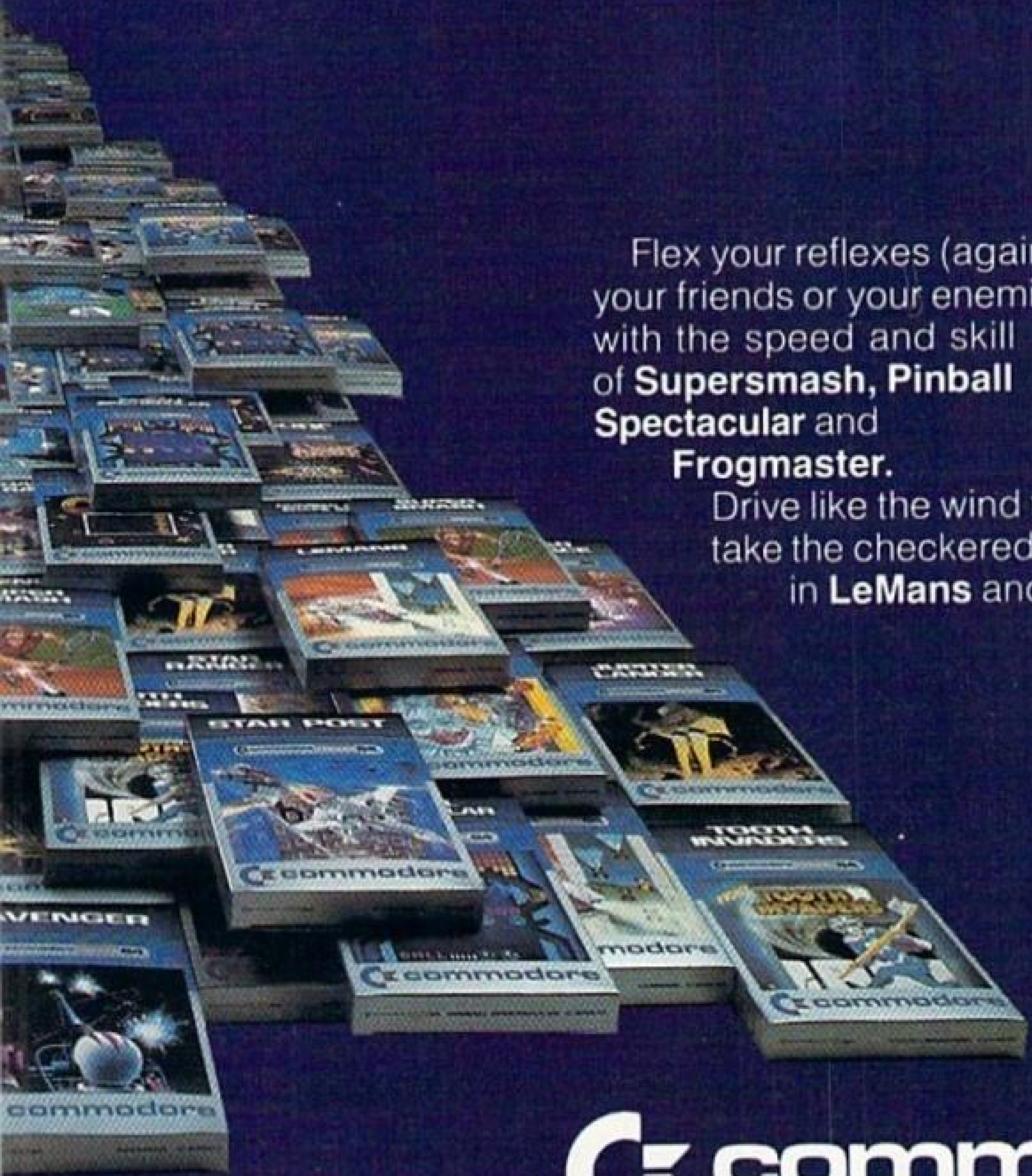
While computers can educate, organize, create and integrate, they can also be fun... with the best games in town.

Commodore's inexpensive arcade quality games can keep everyone in the family amused and entertained.

Teach the kids hygiene while they fight the scary **Tooth Invaders.**

The Best Game in Town.




STAR POST
TOOTH INVADERS
AVENGER
commodore
commodore
commodore
commodore

Flex your reflexes (against your friends or your enemies) with the speed and skill of **Supersmash, Pinball Spectacular** and **Frogmaster.**

Drive like the wind to take the checkered flag in **LeMans** and conquer all the elements, just like the 24 hour road race.

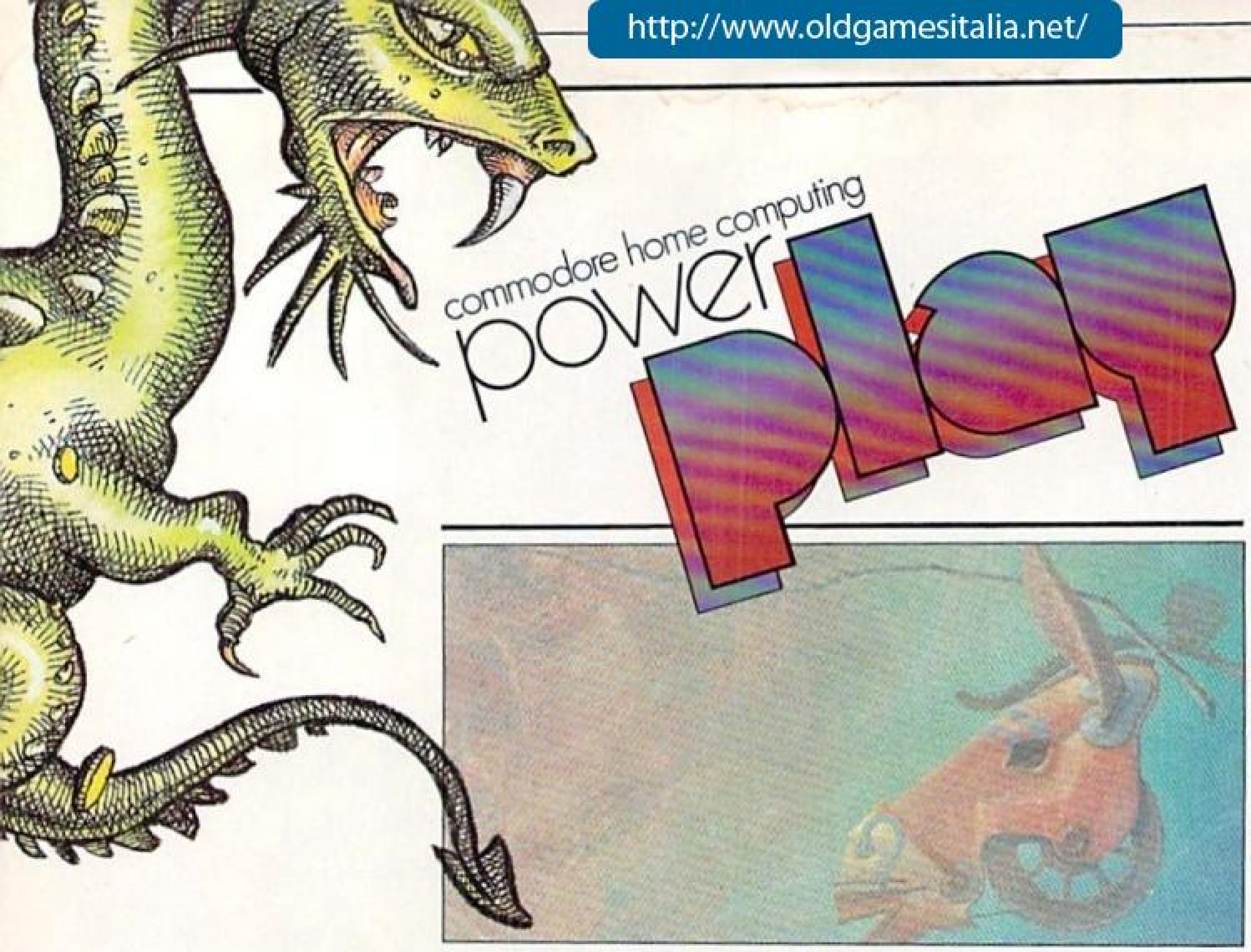
Become a space star as you battle aliens, land your space machine on distant planet surfaces, and defend your base in **Avenger, Jupiter Lander,** and **Star Post.**

commodore
COMPUTERS

First In Quality Software

ur local dealer now . . . He's got the best game in to commodore home computing

power play



features

34 **More Than Just Another Chess Game**
by John Campbell

38 **Can You Survive M.U.L.E.?**
by Neil Harris

45 **For Bridge Players Only**
by Joe Marlino

46 **Suicide Mission on the Russian Front**
by Jim Gracely

50 **Telengard: An Adventure in Terror**
by Cyndie Merten and Bill Hindorff

54 **Archon: Chess, Magic and Monsters**
by Eric and Tom Saberhagen with an introduction by Betsy Byrne, Associate Editor.

departments

8 **Braindrops**
From the editor.

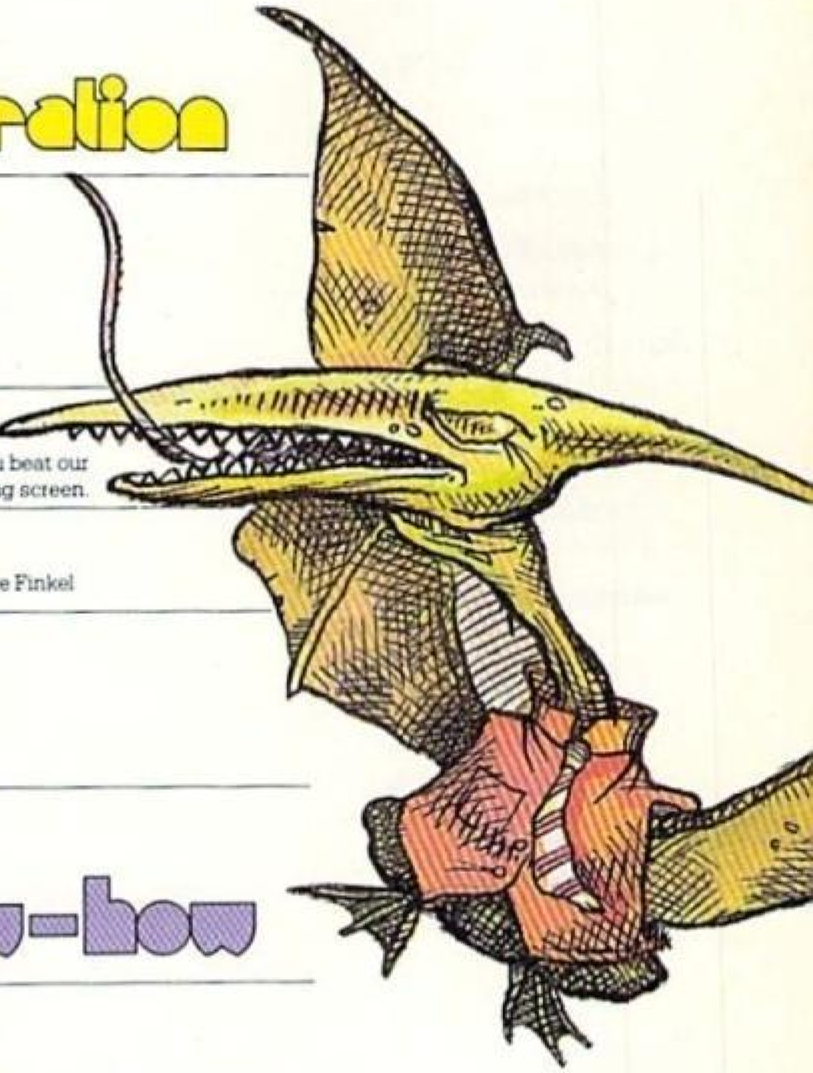
12 **News From the Front**
A directory of computer camps, new games, and a new database for the 64 are in the news this issue.

Volume 3, Number 2, Issue 9 June/July, 1984

18 Book Review
The Commodore Puzzle Book reviewed by Lou Hinshaw

19 Commodore Challenge
Type and save this month's winner. Then fill out the entry blank and send a copy of your best for the VIC 20. Maybe you'll be our next winner!

123 User Groups
A complete list of Commodore user groups around the world.



games & recreation

10 Game Reviews
Beach Head reviewed by John O'Brien
B.C.'s Quest for Tires reviewed by John O'Brien
Blue Max reviewed by Tom McNamara
Trains reviewed by S.G. Grodin

24 High Scores
How do you measure up in our ongoing competition? If you beat our champion gamesters' scores, send in a photo of the winning screen.

22 The Joystick Lunatic
The Fantastic Computer Olympics by Steve Finkel

25 Game Programs
Sky Catcher by Len Lindsay
Houseplant by Marilyn Sallee
Shutter Bug by C.D. Lane

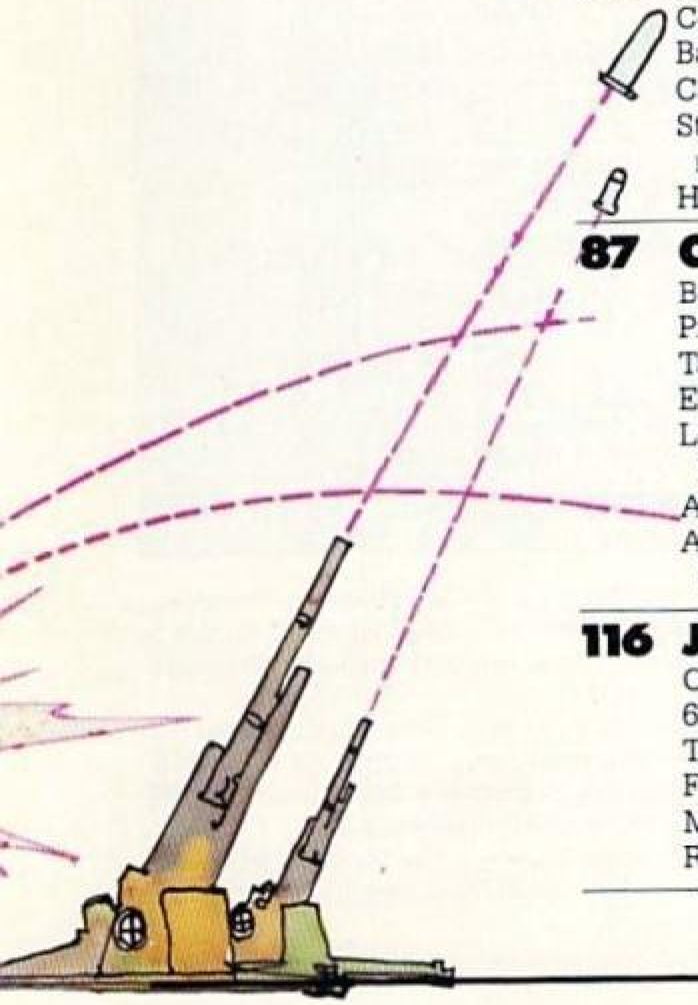
computer know-how

71 No More Pencils . . .
LOGO Hangman by David Malmberg

74 Kids' Corner
Commodore 64 Sprite Contest Winners by Betsy Byrne
Backwords! by Dr. Petrie Curryfavor
Catapult: A Game for the VIC 20 by Joseph Bedard
Student Software: Who Owns What a Student Creates? by Alfred Fant, Jr.
How to Submit Things to Kids' Corner

87 Computer Tutor
Borderline and Type Setter by Richard Stare
Print Gothic or Bold on the 1525 by Brent Miller
Tape Tips by Alfred Fant, Jr.
Editing Characters on the 64 by Keith Golden
Letters to my Grandchildren: Part 1, Roman-to-Arabic and Back by FH Shedd
Automatic VIC 20 Disk Starter by Stephen Leven
All You Need to Know to Begin Programming in Machine Language by John J. Karcher

116 Jiffies
Color Test for the Commodore 64 by Ross Cherednik
64 Typer by Judy Runge
Tape Images on the 64 by Elizabeth Deal
Fun With REM by John Stilwell
More Boolean Foolin' by Marilyn Sallee
Reading the Commodore 64 Keyboard by Ray Sidney



staff

Director/Publishing
Neil Harris

Editor
Diane LeBold

Technical Editor
Jim Gracely

Associate Editor
Betsy Byrne

Staff Writers
John Campbell, Steve Finkel, Joe Marlino, Tom McNamara, John O'Brien

Contributing Writers
Joseph Bedard, Ross Cherednik, Dr. Petrie Curryfavor, Elizabeth Deal, Alfred Fant, Jr., Keith Golden, S.G. Grodin, Bill Hindorff, Lou Hinshaw, Eddie Johnson, John J. Karcher, C.D. Lane, Stephen Leven, Len Lindsay, David Malmberg, Cyndie Merten, Brent Miller, Judy Runge, Eric and Tom Saberhagen, Marilyn Sallee, F.H. Shedd, Ray Sidney, Richard Stare, John Stilwell

Technical Staff
Jeff Bruette, John Campbell, Tony Caramanico, Rick Cotton, Andy Finkel, Barbara Karpinski

Advertising Sales Manager
Pamela S. Fedor

Advertising Coordinator
Sharon Steinhofer

Circulation Manager
John O'Brien

Circulation Assistant
Kathy Reigel

Graphic Design
Neumann Greenberg Schlenker, King of Prussia, Pennsylvania

Cover
Verlin Miller

Printing
Volkmuth Printers
St. Cloud, Minnesota

Typography
Associates International, Inc.
Wilmington, Delaware

Coming Next in Commodore's User Magazines

Commodore Microcomputers, Issue 30, July/August: Our next issue will feature telecommunications—what it is, who's using it and for what, and how you can gain from tuning into the many services available via your computer and telephone.

Power/Play, Issue 10, August/September: We're spotlighting user groups in our next issue, so be sure you don't miss this one. Commodore's user groups are an important force in helping you learn more about your computer, so if you don't already belong to one, you'll find out why you should. Or why you should start one.

Key to Entering Program Listings

```
"[F1,F2,F3,F4,F5,F6,F7,F8]":F1,F2,F3,F4,
  F5,F6, F7 AND F8
"[POUND]":ENGLISH POUND
"[PI]"PI SYMBOL
"^":UP ARROW
"[HOME]":UNSHIFTED CLR/HOME
"[CLEAR]":SHIFTED CLR/HOME
"[RVS]":REVERSE ON
"[RVOFF]":REVERSE OFF
"[BLACK,WHITE,RED,CYAN,MAGENTA,GREEN,BLUE,
  YELLOW]" THE 8 CTRL KEY COLORS
"[ORANGE,BROWN,L. RED,GRAY 1,GRAY 2,L.
  GREEN,L. BLUE, GRAY 3]":THE 8
  COMMODORE KEY COLORS (ONLY ON THE 64)
GRAPHIC SYMBOLS WILL BE REPRESENTED AS
  EITHER THE LETTERS SHFT (SHIFT KEY) AND
  A KEY:"[SHFT Q,SHFT K,SHFT V,SHFT T,
  SHFT L]"
  OR THE LETTERS CMDR (COMMODORE KEY) AND
  A KEY:"[CMDR Q,CMDR H,CMDR S,CMDR N,
  CMDR O]"
IF A SYMBOL IS REPEATED, THE NUMBER OF
  REPITITIONS WILL BE DIRECTLY AFTER THE
  KEY AND BEFORE THE COMMA:"[SPACE3,
  SHFT S4,CMDR M2]"
```

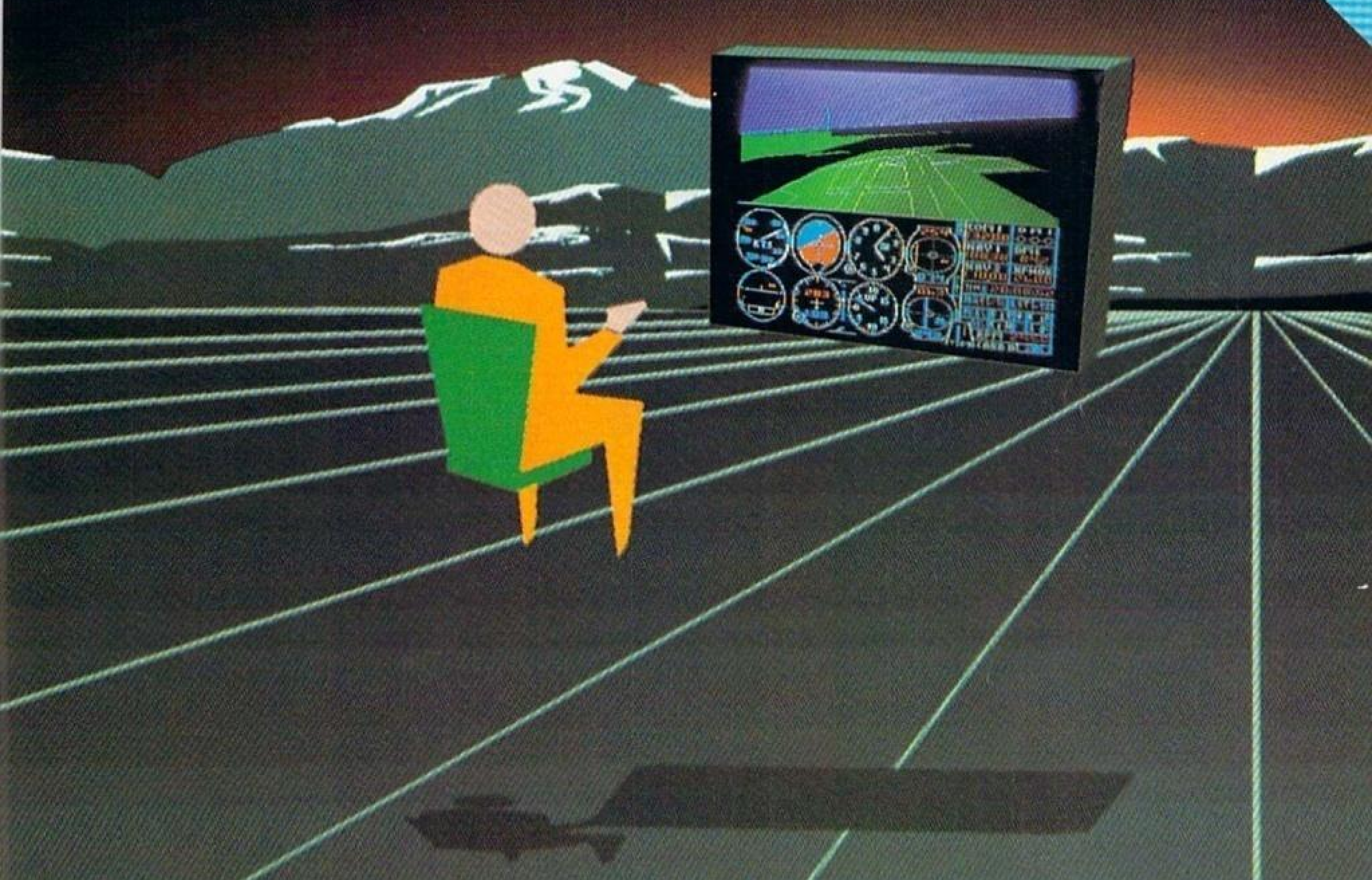
Power/Play is published six times a year by the Computer Systems Division, Commodore Business Machines, Inc., 1200 Wilson Drive, West Chester, PA 19380. Copyright © 1984 by Commodore Electronics Ltd. No material may be reprinted without permission. Volume 3, Number 2, Issue 9, June/July, 1984. ISBN 0-88731-009-5.

Subscription Information: U.S. subscriber rate is $15.00 a year. Canadian subscriber rate is $20.00 a year. Overseas is $25.00 a year. Questions concerning subscriptions should be directed to Commodore Business Machines, Magazine Subscription Department, Box 651, Holmes, Pa. 19043. Phone 800-345-8112. (Pennsylvania phone 800-662-2444.)

VIC 20™, Commodore 64™, and SuperPET™ are trademarks of Commodore Electronics Ltd., PET® is a registered trademark of Commodore Business Machines, Inc. CBM® is a registered trademark of Commodore Electronics Ltd.

Flight Simulator II

For the Commodore 64™



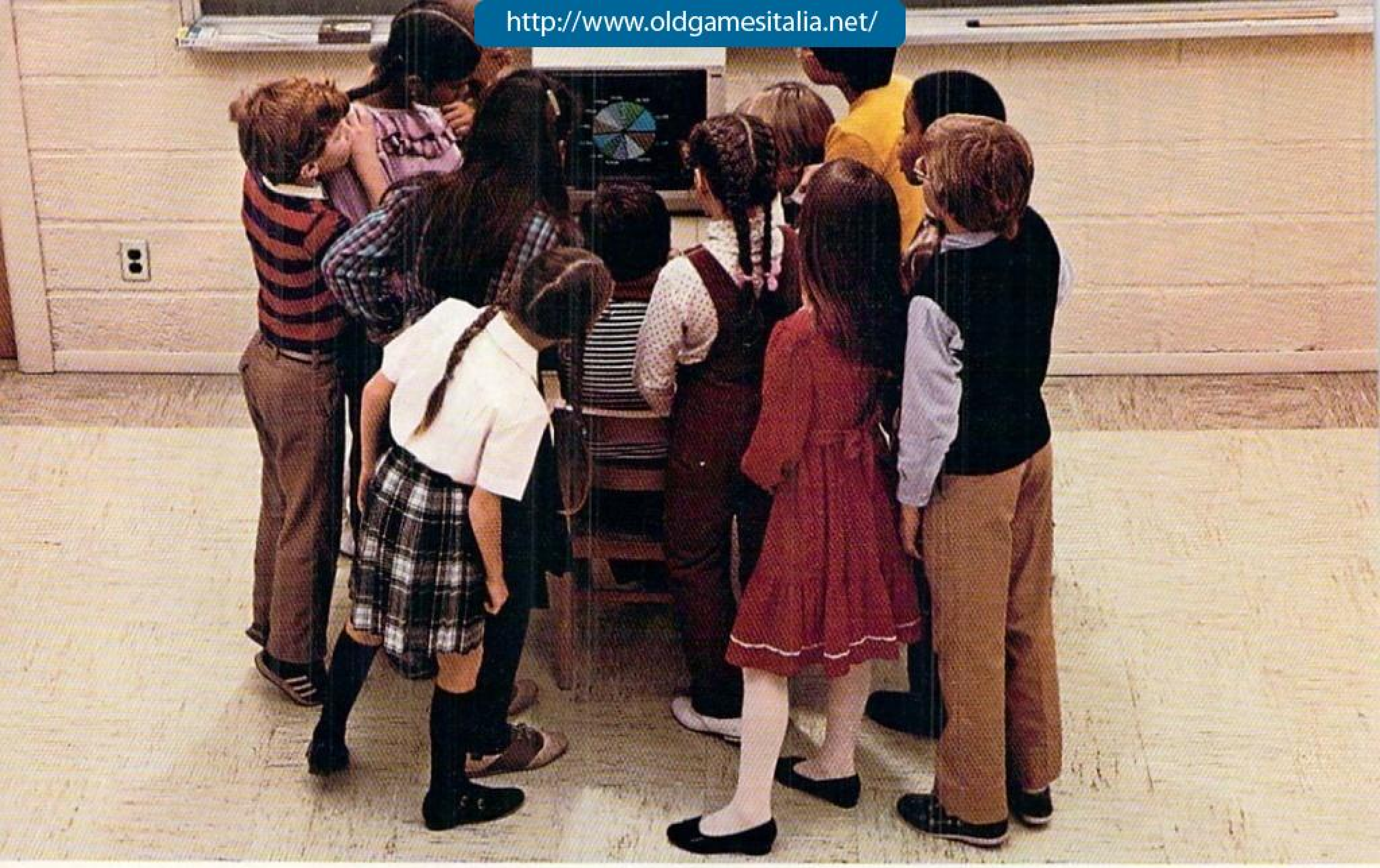
Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

See your dealer . . .

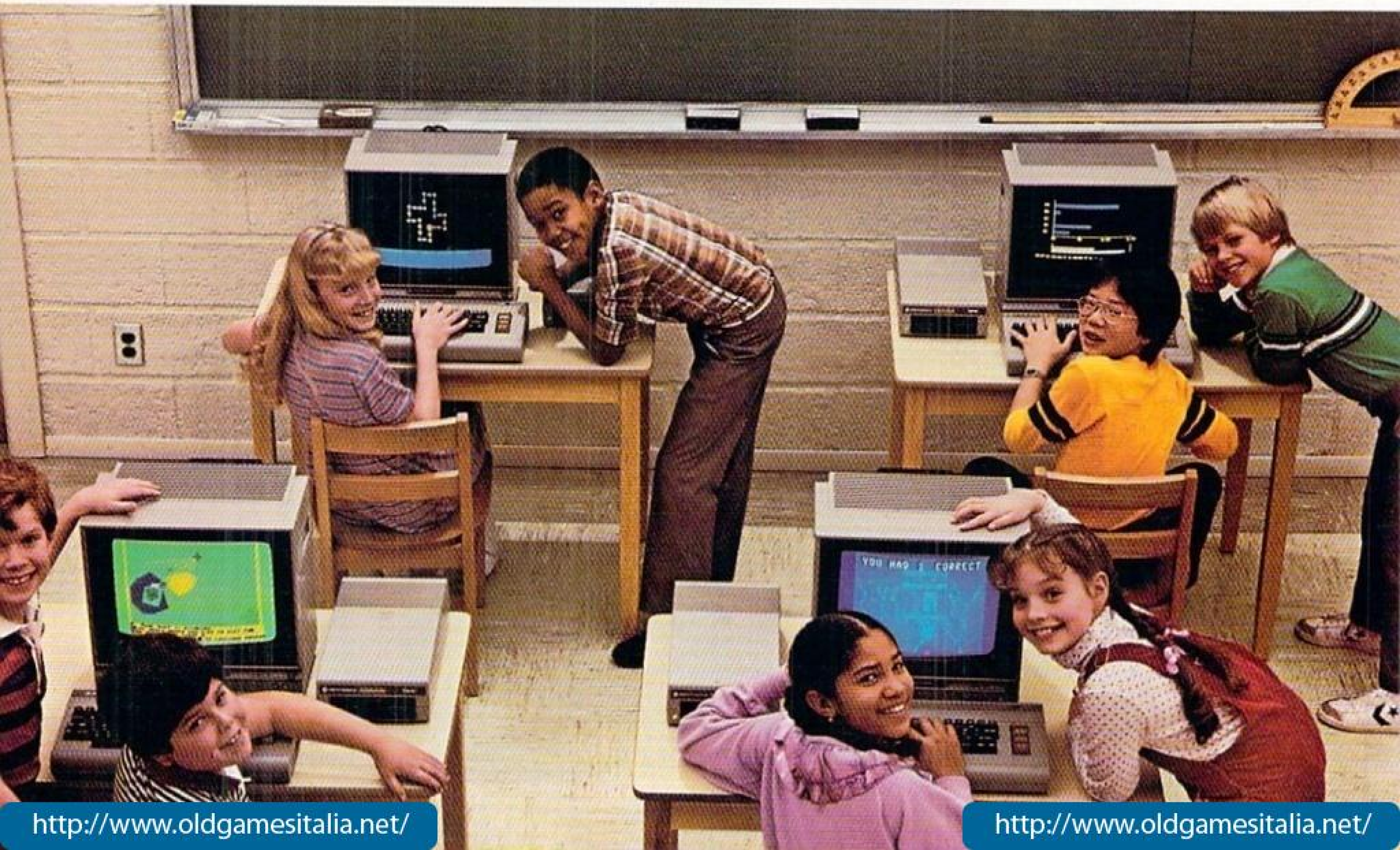
or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995



WHY THE COMMODORE 64 ISN'T IN A CLASS BY ITSELF.


YOU HAD 1 CORRECT

At Commodore, we think it's easier for school children to learn about a computer by using it rather than by waiting to use it.

So, we sell the Commodore 64™ at about half the cost of comparable computers.

With the money you save on the Commodore 64, you can afford the things you'll really need: more Commodore 64's.

In fact you can create a powerful but economical "Local Network" with 8 computers sharing one disc drive.

The Commodore 64 features: 64K memory, 66 key typewriter-style keyboard, 16 color high resolution graphics, 9 octave music synthesizer and 3-dimensional sprites.

And the same commitment we make to hardware, we're making to software. We have highly rated Logo and PILOT programs. Much of the well recognized MECC™ courseware and the Edufun™ Series from Milliken will soon be available. There are hundreds of other programs, including a wealth of public domain software for the Commodore 64. Our newest additions are 30 early learning programs from Midwest Software.

So you see, the all purpose Commodore 64 really is in a class by itself.

For further information on the Commodore 64 and our 250 Educational Resource Centers, contact your nearest Commodore Education Dealer.

A
B
C
D
E
0
5
10
15
OPERATION?(+,-,S)
commodore

commodore
COMPUTERS

Commodore Business Machines Inc., P.O. Box 500M, Conshohocken, PA 19428. Canada—3370 Pharmacy Avenue, Agincourt, Ont. Can. M1W2K4.
d MECC are trademarks of Milliken Publishing Company and Minnesota Educational Computing Consortium respectively.

NEW! IMPROVED!

Here it is, our first bi-monthly edition of *Power/Play.* (By now you've also probably noticed we're using a new format, as well.) We're delighted to bring you more of what you want more frequently in a format we think is easier to follow and even more enjoyable to look at than ever. If you have an opinion, one way or the other, about what we're doing, we'd like to hear from you. After all, you're the reason we're doing it in the first place.

In addition to all the other changes we've been going through in the past few months, we're also expanding our staff. As of this writing, we've added Pamela Fedor as Advertising Sales Manager, but by the time you're reading this we'll have a new assistant editor, who'll be handling news and reviews, and a programming assistant who'll keep Jim Gracely, our technical editor, from drowning in disks and program listings. (I can still see the top of his head, so I know he's in there somewhere.)

Recently we got a list of the top twenty best selling computer books from B. Dalton Booksellers, a national bookstore chain with over 700 stores. The list surprised even me. In that top twenty, thirteen—count 'em—thirteen—were books about the Commodore 64. This means people are not only *buying* Commodore computers, they're also learning to *use* them. Which reminds me to remind you that if you want to get the most out of your computer, you have many resources—including a whole raft of books devoted to your specific computer, be it a Commodore 64, VIC 20 or PET/CBM.

In this issue we're featuring the best of the new strategy games for the Commodore 64. (You'll also notice that our Circulation Manager, John O'Brien, has taken out his old word processor and polished it up again after many months of disuse, to produce a couple of game reviews.) And if you're a VIC 20 owner you'll find a lot of information about your computer in our various departments. In fact, if you turn to Kids' Corner you'll find yet another dubious contribution from the mysterious Dr. Petrie Curryfavor—a second "un-word" processor for creating more secret messages with your VIC.

However, if you want to use your VIC more "seriously" (oh come now), you'll like Richard Stare's "Borderline" program even better. Or perhaps having your 1525 printer learn to speak gothic and bold will appeal to you. My point is that whatever your interest, you'll find something in this issue for you. Provided you're interested in having fun with your computer, that is.

If you're wondering what's new in software for your Commodore computer, we can assure you a LOT is new, judging from the overwhelming quantity that was shown at the Softcon software trade show in New Orleans this past February (you have to realize that I'm writing this in March, so this is still news). According to our reliable sources, the most outstanding packages shown by independent manufacturers were Waveform's *MusiCalc* and keyboard for the Commodore 64, *Moviemaker* for the Commodore 64 from Interactive Picture Systems, the "Tymac Talkies" series for the VIC and 64 (from Tymac, of course—a company that, I hear, operates out of a converted A & P supermarket in Franklin, New Jersey), the new Reader's Digest series and an educational series from a new company called Mindscape. Look for reviews of these and other products in upcoming issues.

Before I close it up for this month, I think I should mention a new book from Osborne/McGraw-Hill titled *Home Computer Software Guide* by Steve Ditlea. Normally, O/McG-H produces high quality books that you can rely on for good information, but it looks like somebody wasn't paying attention when this one came through. Although its copyright date is 1984, this so-called "guide" doesn't even mention the *existence* of either the Commodore 64 or PET/CBM computers, let alone talk about software for them. And it provides only a sketchy (I'm being kind) representation of what's available for the VIC 20, although I suppose we should be grateful that the author at least noticed the VIC is around here and there. We don't recommend it. **C**

Diane LeBold

LONDON BLITZ

London Blitz

The streets of London are threatened with deadly V-1 rockets. You, as a member of Her Majesty's Royal Bomb Squad, must disarm each one before its timed fuse detonates. A variety of bombs with infinite defusing combinations make for endless nail-biting excitement.

A combination of logic, skill and a little luck is required in this highly-acclaimed computer masterpiece.

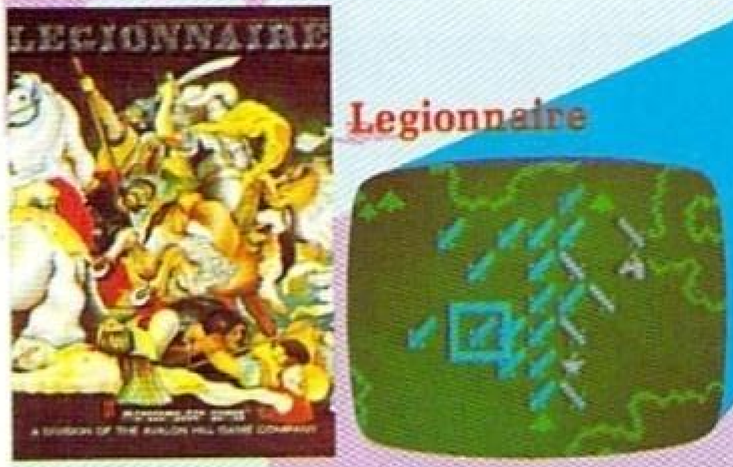
Cassette for Commodore 64*, $20
Disk for Commodore 64*, $25
Playing time: 20 minutes to 3 hours;
Intermediate complexity level

Panzer-Jagd

The endless Russian steppe trembles again wi the rumble of invading panzers, and this time you are in command! Your units include platoons of Panthers and PzKw III's with infantry support, all maneuvering over an ever-changing battlefield map. Off-map artillery support is also available to help you combat the hordes of Russian units.

Not just an arcade shoot'em up, **Panzer-Jagd** requires careful tactical planning and, above all, a determination to win. Hi-res graphics and sound effects.

Cassette for Atari* (32K) and Commodore 64*, $25
48K Disk for Atari* Home Computers, $30
64K Disk for Commodore 64*, $30
Playing time: 1-4 hours; Intermediate complexity level



Legionnaire

The beat of barbarian war drums echoes through the valleys of Gaul; the crack legions of Imperial Rome are on the move. Outnumbered two-to-one by multitudes of heathen infantry and cavalry, you, as Caesar, must select your terrain and tactics carefully or face annihilation. Beautifully detailed scrolling map lets you examine the entire battlefield in this realtime game of ancient warfare.

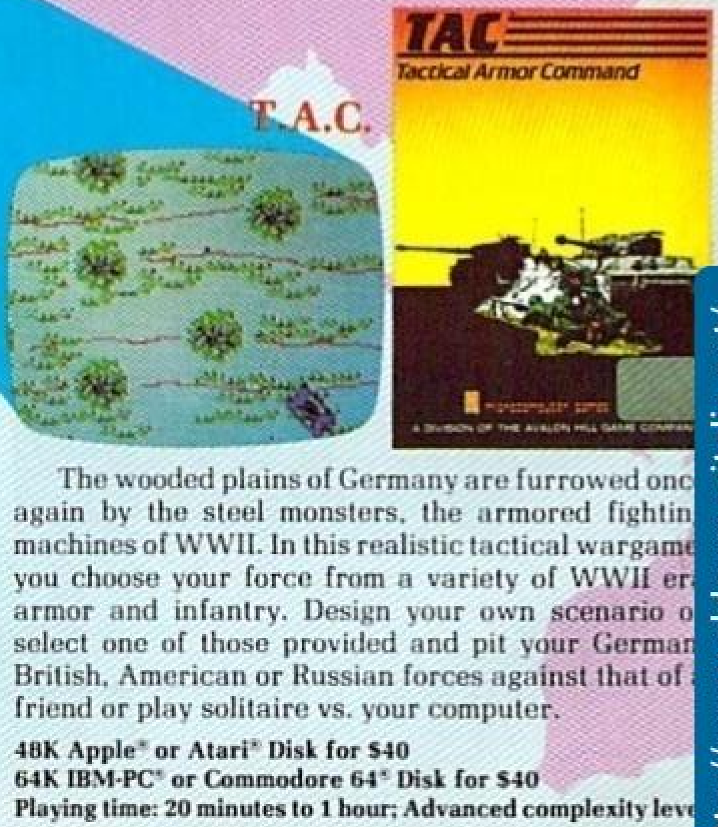
Cassette for Atari* (16K) and Commodore 64*, $35
48K Disk for Apple* II, II + & IIe, $40
32K Disk for Atari* Home Computers, $40
64K Disk for Commodore 64*, $40
Playing time: 20 to 45 minutes; Intermediate complexity level

A World of Games

microcomputer games®
A Division of The AVALON HILL Game Company
4517 Harford Road • Baltimore, Maryland 21214

These and many other fine Avalon Hill
Microcomputer Games are available
at leading computer game stores everywhere . . .
or call Toll-free: **1 (800) 638-9292** for the name of a
store near you. Ask for Operator **W1.**

* Trademarks of Apple Computers, Inc., Warner Communications, Commodore Business Machines and International Business Machines.



T.A.C.

The wooded plains of Germany are furrowed onc again by the steel monsters, the armored fightin machines of WWII. In this realistic tactical wargame you choose your force from a variety of WWII er armor and infantry. Design your own scenario o select one of those provided and pit your German British, American or Russian forces against that of friend or play solitaire vs. your computer.

48K Apple* or Atari* Disk for $40
64K IBM-PC* or Commodore 64* Disk for $40
Playing time: 20 minutes to 1 hour; Advanced complexity leve

game reviews

B.C.'s Quest for Tires

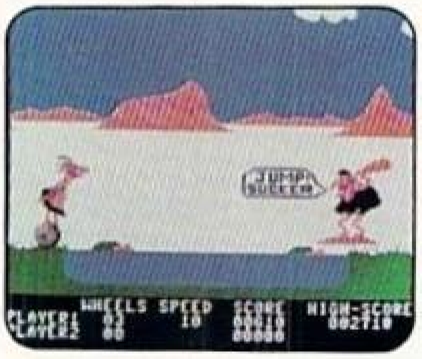
Reviewed by John O'Brien
Commodore Publishing

Sierra On-Line
Coarsegold, CA 93614

This game on disk for the 64 is based on the famous comic strip by Johnny Hart. The player controls Thor on his famous wheel. Thor can only jump or duck and speed up or slow down and the player can control the length of these motions. The only other skill needed is timing.

The object of the game is to rescue the pretty girl who yells "help help" throughout the game. It starts out easy enough, you just roll along jumping over some ruts and bumps in the road. But then the game gets progressively harder with rolling rocks, ponds, caves, dinosaurs and even "the fat broad" (don't send the nasty letters to me—that is what they call her in the comic strip) coming between you and the damsel in distress.

The game does have some excellent qualities. The graphics and scrolling action are very good and entertaining as well. The characters are true to the comic strip and Thor gives you a perturbed look when he is sinking in the pond. The level of difficulty can be changed by speeding things up. And although there are a few difficult moments, they become easy once you figure them out.



One of the difficult moments occurs when you get to the first pond. Turtles are moving up and down in the pond and "the fat broad" is waiting with a club at the other side taunting you with "jump sucker." The trick here is to wait for the moment when the turtles all come up at the same time. It takes a while, so be patient. They always come up in the same pattern, so you only have to figure it out once.

B.C. may be scoffed at by the serious gamer (if a gamer can indeed be called serious) for the limited challenge of the play action. This same feature makes it most enjoyable for younger game players.

B.C.'s QUEST FOR TIRES
TM
Sierra ON-LINE TM

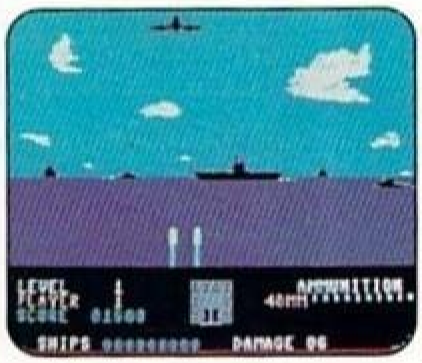
Soft War

Beach Head

Reviewed by John O'Brien
Commodore Publishing

Access Software
925 East 900 St.
Salt Lake City, UT 84105

In *Beach Head*, the player's mission is to destroy the enemy's big gun. However, there are a few things that stand in your way, including torpedo-infested waters and a heavily armed enemy fleet. Obviously, *Beach Head* will be enjoyed most by war simulation enthusiasts, but many other game players will also like the exciting play action and strategies needed to win.



The best thing about *Beach Head* is that there are five screens to master and they all require different skills and strategies. On the torpedo level, for example, you must remember that you can steer only left or right. The up and down motion of the joystick will only speed your ship up or slow it down.

The torpedo screen is optional, which brings up another interesting feature of the game. There are two basic strategies the player can choose from. You can either engage the enemy directly or try a sneak attack using the dangerous secret passageway. It may seem easier at first to use the direct attack, because the secret passage is where you encounter the torpedoes. But I found the sneak attack a much better strategy, even though you can suffer heavy casualties trying to get your ships through when you first try it. It takes time to get good at this screen but it is worth it because if you use this strategy, the next two screens are much easier.

On the next screen you encounter the enemy fleet. You are bombarded by planes coming off the enemy aircraft carrier. If you don't chose the sneak attack, this screen can be a real wrist buster, because you have to shoot down so many planes. Rapid fire is the best strategy. This screen features the best graphics and sound effects of any screen in the game. The planes look and sound as if they are really firing on you and then rolling out of range of your gun. The planes must hit you several times before they destroy a ship so keep your cool and keep firing. If you survive this round there are only five ships between you and the beach.

The graphics and play action on the beach screen are a little disappointing. The graphics are flat and the play action is limited, because you can only maneuver your tank up and down; the speed cannot be controlled. Despite this the screen is challenging.

I didn't like the fact that once you knock out the big gun, the game ends. So you can "win" the game with a relatively low score. But once you get good at the game you're going to want to pick up as many bonus points as possible to get a high score. The higher skill levels just seem to speed things up.

Overall the game promises hours of fun. There are four skill levels and up to two players can play at once.

Blue Max

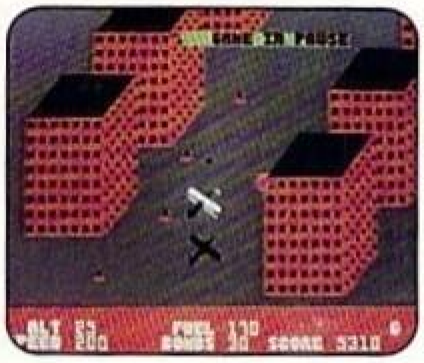
Reviewed by
Tom McNamara
Commodore Advertising

Synapse Software
5221 Central Avenue
Richmond, CA 94804

Did you think that *Blue Max* is your typical aerial combat game? You're wrong! *Blue Max* proves itself to be one of the hottest games so far for the Commodore 64, with some very entertaining arcade features.

As a World War I flying ace, your mission is to attack the enemy by shooting, bombing and strafing everything in enemy territory. The targets include enemy planes, buildings, bridges, tanks and ships. In order to gain entry to the next level you must hit a certain number of the primary targets. The ultimate goal is to safely get into the enemy city and successfully knock out the three special targets.

Blue Max is a three-dimensional diagonal scrolling game which at first seems graphically reminiscent of *Zaxxon*. Besides keeping an eye on the instrument panel for altitude, fuel, speed and bombs, the player must watch out for planes that attack from both ends and also anti-aircraft firing from the ground. This is what makes the game so frustratingly entertaining.



What I like best about *Blue Max* is the play action. It has a great simulation and feel for an airplane in flight. For instance, when it's windy or when the plane is damaged due to anti-aircraft, the plane loses its maneuverability. The instrumentation panel is also easy to understand without being distracting. Using the shadow of the airplane gives the player a great feel for altitude and bombing.

One of the nicer features is the pause, which will allow the player to pause the game with no time penalty. You also have the option of setting the joystick for forward to climb, back to descend and vice versa. Using the joystick in the game's reverse mode allows for the standard airplane joystick action—pulling back to climb in altitude and pushing forward to descend.

There are a few things to remember when you're trying to increase game play, such as not bombing your own airbase. This results in a crash and a much shorter game. Another good tip is that if you land as close to the beginning of the runway as possible, you will have plenty of speed for the next take-off. This is important because not enough airspeed during take-off will also result in a crash. I also found out that it is much easier to sink the ships by strafing than by bombing. You will need as many flashing targets as you can get if you are to win at *Blue Max*!

Finally, when you reach the quota of key targets destroyed, you enter the enemy city. The three targets there must be successfully bombed in order to complete the mission and receive the award.

After many hours of play, *Blue Max* maintains its fun and challenge. It is a "must" for the aerial combat game connoisseur as well as strategy game connoisseurs of all ages. Overall, *Blue Max* is a "four star" action game!

Trains

Reviewed by S.G. Grodin
Dallas 64 User's Group

Spinnaker
215 First Street
Cambridge, MA

Spinnaker products, in general, define their target audience well. *Trains* is identified for an audience of ages ten to adult. This writer believes that a strong eight year-old could probably enjoy and learn from at least the first two levels.

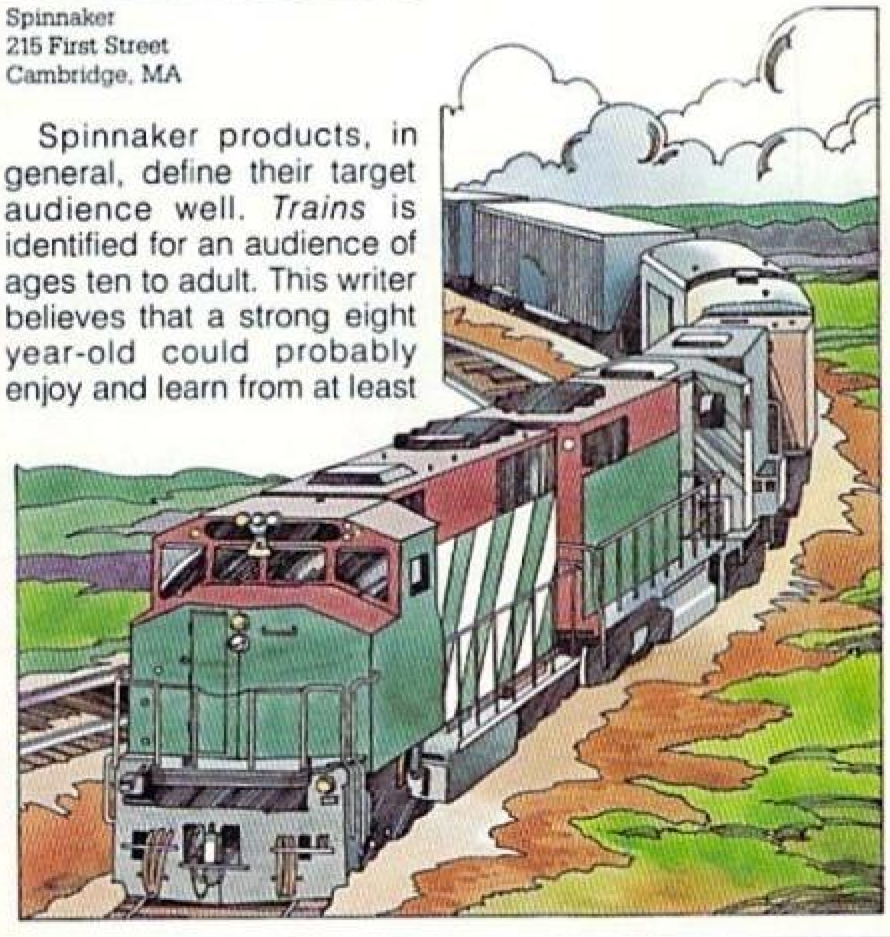
The main objective of *Trains* is to profitably run a railroad. In order to do this, you must bring needed materials to various industries, buy them from various sources, and have enough money to pay for fuel, labor and repairs to continue running your train.

If this sounds terribly familiar, then you'll love it! It takes the next logical step-up for successful *Monopoly* enthusiasts without threatening the train hobbiest with extinction.

Trains has eight levels of play, with each level getting progressively more difficult. The music, sound, color and graphic representations are ranked as "admirable". The joystick control is smooth and faultless. Loading is typically Spinnaker simple and the packaging is first class.

Don't let all these accolades fool you, *Trains* is challenging, frustrating, demanding and enormous fun. In spite of all this rhetoric, I'd be sure to give myself about two light-years of time on Level 1 so as to have a fighting toot of a chance with Level 2.

Trains is a winner and well priced in the $35-$40 range.



Free Computer Camp Directory

The traditional fun and challenge of summer camps now includes the bits and bytes and RAMs and ROMs of computer programming. To help parents and children choose from the growing number of day and resident summer camps that provide computer instruction, Verbatim Corporation, in cooperation with the American Camping Association, is offering a free reference guide, "Camps 'n Computers—A Directory of Summer Computer Camp Instruction in the United States."

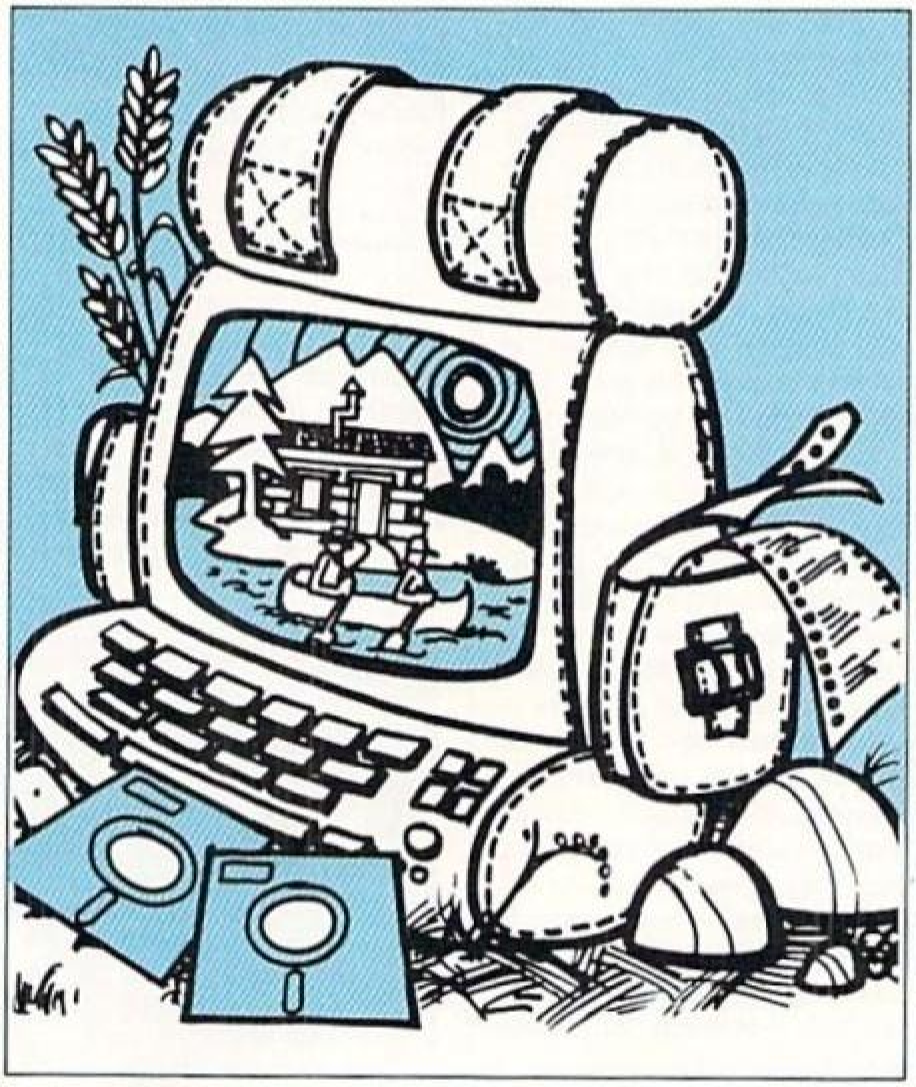
With more than 100 listings, the directory divides the camps into six regions: New England, Middle Atlantic, South, Midwest, Southwest and West. Each entry includes the camp contact, address and telephone number, tuition, length of sessions and recreational facilities. Also included are the number and types of computers, the student/computer ratio, instructor qualifications, computer programs taught and the number of daily computer hours available to campers. Camps with ACA accreditation are clearly identified, with colleges and universities as well as other camps constituting the remaining instructional sites.

"As summer computer camp instruction increases in popularity, we're delighted to assist parents and children with a directory like Camps 'n Computers," says Armand B. Ball, executive director at the ACA. "Its listings will help direct families to the many fine day and resident camps now providing this special kind of instruction."

When deciding on a computer camp, Ball recommends that individuals first determine their own level of computer experience and then use the directory to identify the camps that offer the most appropriate level of instruction. "Some camps teach sophisticated computer languages like Pascal or FORTRAN for experienced computer users, while others specialize in LOGO and BASIC, which are more applicable for beginners," he adds.

Most of the camps also offer a wide variety of recreational and other educational opportunities, combining traditional organized camp activities with computer instruction. "The atmosphere at summer computer camps is ideal for anyone interested in learning about computers," says Shely Saidman, manager of account marketing and special educational projects at Verbatim, a leading manufacturer of magnetic media products. "The camps do a great job at integrating softball with software."

For a copy of the directory, send a check for $1 (for postage and handling) payable to Verbatim, at: "Camps 'n Computers," Suite 228, 4966 El Camino Real, Los Altos, CA 94022.



Use BASIC to Solve Everyday Problems

Creative Computing Press has released a series of six Ideabooks by David H. Ahl. Versions are available for most microcomputers, including the Commodore 64.

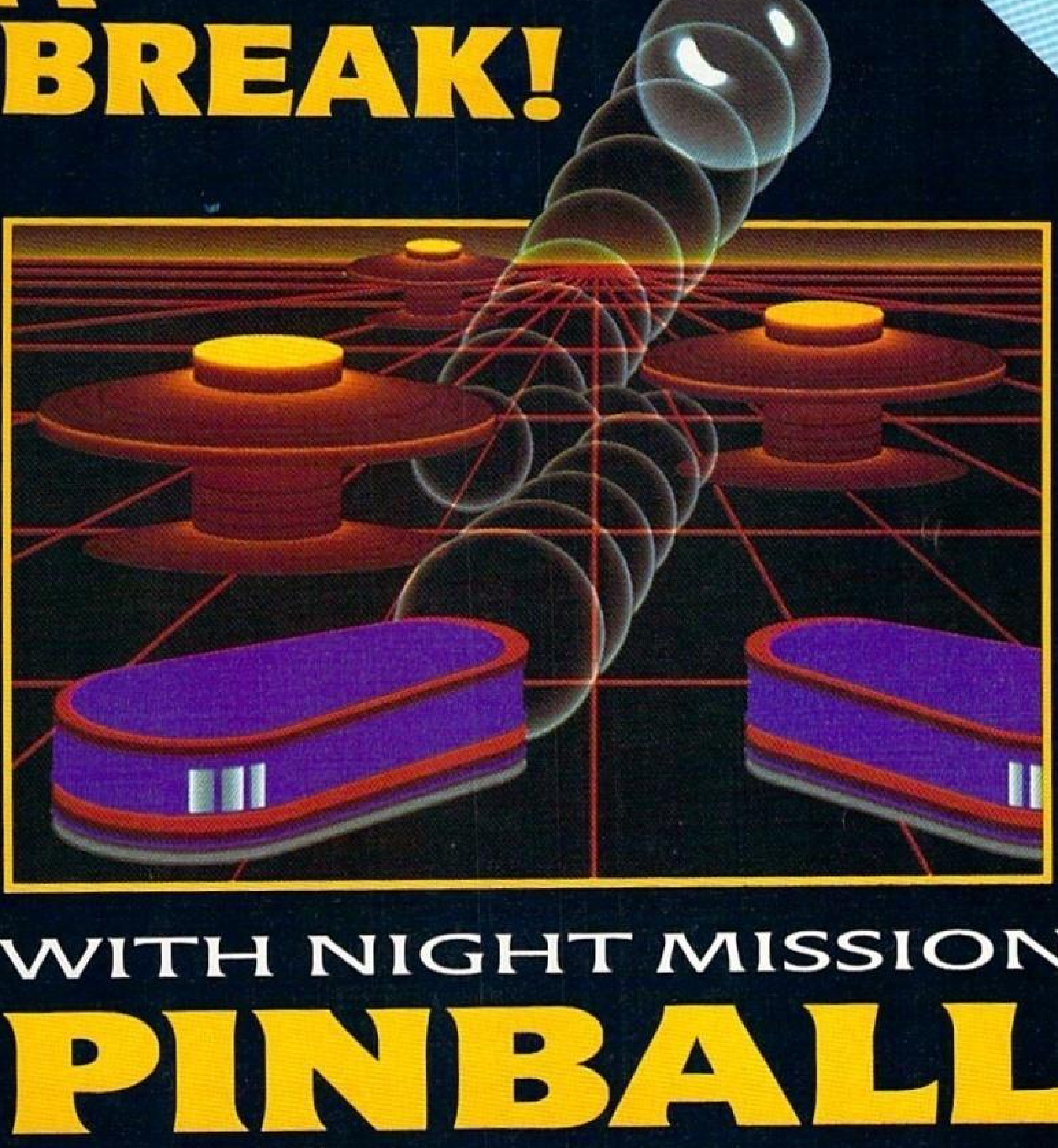
The Ideabooks demonstrate dozens of ways to make the most of a computer for solving both everyday and unusual problems. The 50 ready-to-run programs illustrate scores of different techniques for solving problems in mathematics, science and business.

The ten chapters deal with formulas, repetitive trials, convergence, recursion, compounding, probability, geometry, simulations and drill-and-practice.

The author, David H. Ahl, has been involved with the use of computers since 1957. He is the author of 16 books and is the founder of *Creative Computing* and *SYNC* magazines. The book is delightfully illustrated with drawings and cartoons by Wayne Kanishiro. Softbound, 150 pages, 5½" × 8", $8.95, the books are available from Creative Computing Press, 39 E. Hanover Ave., Morris Plains, NJ 07950.

TAKE A BREAK!

For Commodore 64



WITH NIGHT MISSION
PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of Night Mission PINBALL, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put Night Mission PINBALL in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with Night Mission PINBALL from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.

See your dealer . . .

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820 USA
(217) 359-8482 Telex: 206995

commodore 64

THE BEST GAMES IN TOWN

Arcade Action Games

Pinball Spectacular: Real pinball action and thrills. Sound you won't believe. Chutes, lights, bumpers and more.

Supersmash: Raquetball arcade classic. 3 games in 1. Many skill levels keep the challenge alive.

Tooth Invaders: Reviewed by American Dental Association. Arcade action teaches good dental care. Beat D.K. at all 9 play levels.

NEW **Star Post:** Protect the Star Post from waves of invaders. 3 levels of skill. 99 levels of action.

NEW **Avenger:** Destroy attacking aliens with laser cannons. Classic arcade action. Multi-speed attacks.

Jupiter Lander: Space landing simulation. Horizontal/vertical thrust. Soft-land scoring. Wow! animation.

Radar Rat Race: Beat the maze. Eat all the cheese. Beware deadly cats/rats. Cartoon action fun for all ages.

NEW **Lemans:** Multi-obstacle road racing at its best. Arcade action and graphics. Night, water and divided highway hazards.

NEW **Star Ranger:** Fight your way through hoards of space enemies. Avoid asteroids and land safely. Superb graphics and space action.

NEW **Frogmaster:** Unique sports challenge. Train animals to play football and rugby. Over 100 variations. Play against, computer, friend or yourself.

RADAR RAT RACE
PINBALL SPECTACULAR
FROG MASTER
JUPITER LANDER
TOOTH INVADERS
STAR POST
STAR RANGER
LEMANS
commodore

Children's Series

Introduction to Basic I: Simple step-by-step instructions. Modular design. Practical BASIC applications as you learn.

Zortek and the Microchips: Award winning program teaches children BASIC through games, graphics and stories.

Easy Lesson/Easy Quiz: Take the drudgery out of writing tests and quizzes. Answer keys provided. 7 categories per test.

NEW **Number Nabber/Shape Grabber:** 2 Learning games in 1. Build both math and object identification skills. Lively graphic and sound effects.

Visible Solar System: Fly the solar system. Land on planets. Calculate age and weight. Astronomy for home and school. Award winner.

Speed/Bingo Math: 2 games in 1 teach children 4 to 10 basic math skills. Beat the clock or your friends.

SPEED/BINGO MATH
VISIBLE SOLAR SYSTEM
EASY LESSON EASY QUIZ
ZORTEK AND THE MICROCHIPS
INTRODUCTION TO BASIC/PART I
NUMBER NABBER SHAPE GRABBER
commodore

Bally Midway

NEW **Gorf:** 4 Space action games in 1. Fly your fighter defeat "The Empire". Multi-skill levels. IT TALKS! (with Magic Voice)

NEW **Wizard of WOR:** Fight your way through 30+ mazes. Defeat the Wizard and Worriors. Multi-skill. IT TALKS! (with Magic Voice.) Award winning conversion.

Seawolf: The classic battle at sea. Destroy PT Boats and Destroyers. Great graphics and sound.

Omega Race: Fast space race action. Many skill levels. Avoid deadly mines as you eliminate droid forces.

Clowns: Amazing action under the "Big Top". Help clowns "pop" balloons. Colorful acrobatics. Fun for all.

Kickman: Ride the unicycle and catch falling objects. Multi-skill levels. Tuneful sound. Watch out! Don't fall!

NEW **Blueprint:** Help J.J. build the "Ammo Machine". Parts are stored in a colorful maze of houses. Multi-skill and difficulty levels.

NEW **Lazarian:** 4 different screens. Multi-skill level space action. Rescue, evade obstacles and destroy a one-eyed leviathan.

WIZARD OF WOR
KICKMAN
BLUEPRINT
GORF
CLOWNS
LAZARIAN
OMEGA RACE
SEA WOLF
commodore

Adventure Games

Zork I: Fantasy adventure in a dungeon. Find all the treasure and escape alive.

Zork II: This dungeon adventure dares you to find treasure and secret places and still survive.

Zork III: The ultimate dungeon test. Discover the Dungeon Master's secret purpose and come out alive.

Suspended: Awake in 500 years. Solve varied real and original puzzles to save the planet from total destruction.

Starcross: Travel through the mystery ship. Meet aliens friend and foe. Face the challenge of your destiny. Map of galaxy included.

Deadline: Find the murderer and solve the mystery all in 12 hours. Inspector casebook and evidence included.

ZORK I
STARCROSS
DEADLINE
SUSPENDED
ZORK III
commodore

Music Series

Music Machine: Play piano or organ melodies and percussion rhythms together. Music staff shows notes on screen. Vibrato, tempo and pitch controls.

Music Composer: Create, play and save your tunes easily. Simulates up to 9 instruments. Notes appear on screen. Play your keyboard like a piano.

MUSIC MACHINE
MUSIC COMPOSER
commodore

commodore
COMPUTERS

First In Quality Software

Commodore 64
SLALOM!
SLALOM!
TRONIX

Commodore 64
WATERLINE
WATERLINE
TRONIX

Commodore 64
SUICIDE STRIKE
SUICIDE STRIKE
TRONIX

Four New 64 Games

Four new Commodore 64 games featuring dramatic action and sophisticated strategies are now available from Tronix Publishing, Inc.

The two new action games, *Slalom* and *Motocross*, and two adventure strategy games, *Waterline* and *Suicide Strike*, join the successful *Kid Grid* and *Juice!*—other Tronix games for the Commodore 64.

Tronix's adventure strategy games present the player with tough choices and hair-raising challenges. For example, in *Waterline*, created by Steve Morris, the player can be a hero and save the ship's passengers. He can rescue the ship's valuable cargo by grabbing a few bars of gold.

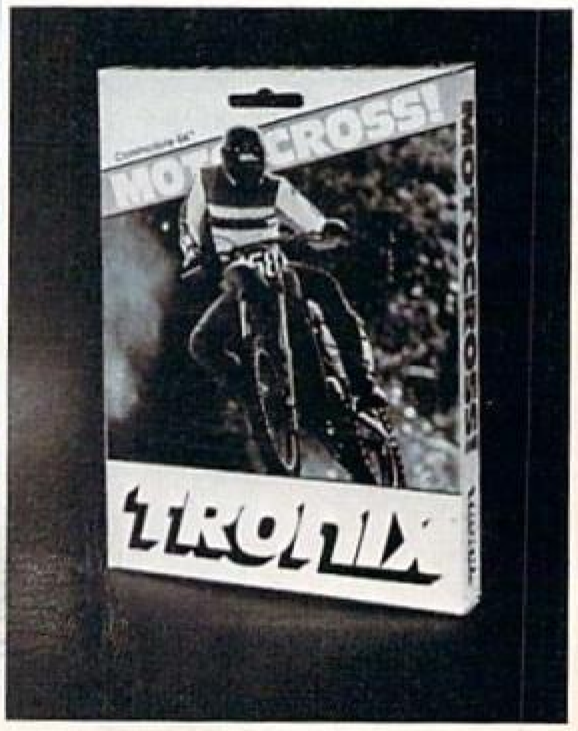
In *Suicide Strike*, designed by Graham McKay, the player is challenged to maneuver his aircraft through waves of enemy planes and gun-fire to reach his military target. A shrinking fuel supply and a tight deadline for returning to his home base add tension and suspense to the player's strategic moves.

One of the most notable features of *Suicide Strike* is the game's rear-view mirror, which shows action behind as well as in front of the player. It works just like a real rear-view mirror, with planes flying in towards the player. If the player maneuvers to the right, approaching planes appear to move left. The player has to stay on top of three things: approaching enemy fire, a diminishing fuel supply, and a clock running out of time.

Tronix president John Reese said that computer gamers have become so sophisticated that new offerings have to involve the player at more levels in order to sustain the elements of interest and fun.

"Most of the early games were pretty straightforward," he said. "But today's players want refined characters, play, packaging and animation."

More information on Tronix products is available at the company's headquarters, 8295 S. La Cienega Blvd., Inglewood, Calif. 90301. The telephone is (213) 215-0529.



AdventureWriter Games Systems Programmed in English

Dynatech Microsoftware, Inc., has introduced AdventureWriter™, a games system based on their CodeWriter™ concept, that allows the user to design games by programming in English language.

"AdventureWriter provides the software owner with some significant advantages over purchasers of packaged games that are "locked" in a single format," said Warren Shore, President, Dynatech Microsoftware.

"One distinct advantage of AdventureWriter and all of our CodeWriter-based products is versatility," Shore said. "You are not buying a one-purpose game application. You are getting a games system that allows you to create a virtually limitless number of games."

AdventureWriter also provides another level of entertainment—the challenge and sense of accomplishment of designing your own unique games, Shore added.

AdventureWriter games are easily designed. The software instructs the user how to proceed through a systematic process of building a format, requiring the operator only to type in appropriate responses in simple English.

Throughout the programming process, AdventureWriter provides "open windows" that allow the operator to build new concepts into the game. Treasure hunting, jungle escapes, mazes, hazards, villains and heroes can be designed by the user.

Once the game is set, AdventureWriter automatically converts the English responses into computer language and the game can be recorded on the user's own diskette.

Since each game created is the user's own design, the "author" can claim ownership and even copyright the game.

⁂

Database for the 64

COMPUTERSTUFF announces the release of a multi-functional database program for the Commodore 64.

DATA*MAX is a flexible, easy-to-use database system that can balance your checkbook, keep a household budget, record collections and take care of almost any home filing chore. The software and extensive user guide are written especially for the newcomer to computers. All options are screen prompted and require a single key stroke to execute, eliminating the need for constant referrals to the manual.

DATA*MAX can maintain an unlimited number of 10,000-character files containing up to 200 eight-field records each. A sub-file option allows the user to split a full file into two or more smaller ones, enabling expansion beyond the 200-record limit. A special "escape key" allows any program option to be aborted without loss or change of data. Use of color displays and full sound prompting give instant visual and aural response to user input and error conditions.

DATA*MAX is written in compiled code for fast search, sort and disk access. It requires a Commodore 64 and disk drive. Detailed user guide includes explanations of how to use DATA*MAX as a checkbook/budget manager, address book, insurance inventory and simple accounts receivable/payable recorder.

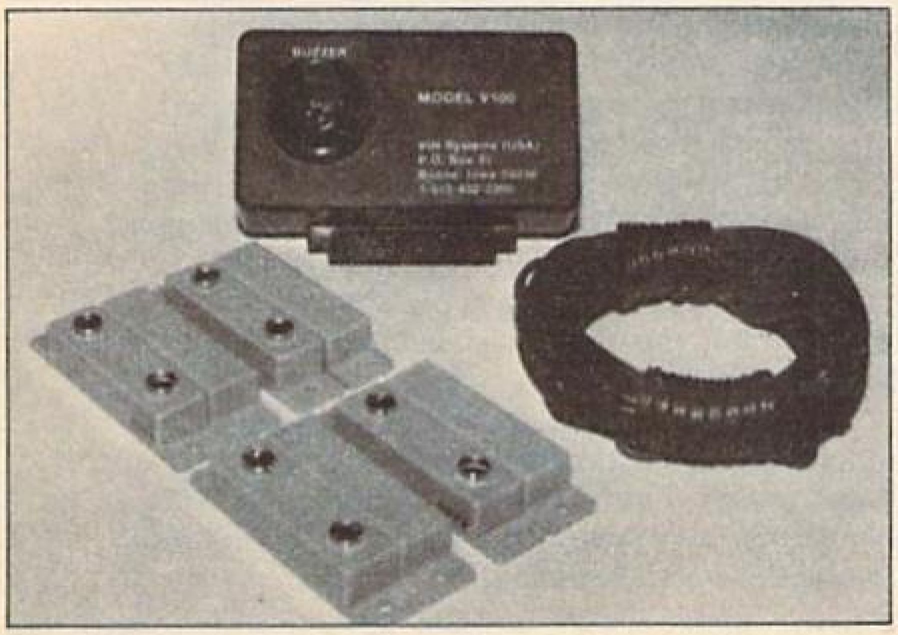
The package is available only for the Commodore 64 on diskette for $29.95 from COMPUTERSTUFF 308-½ Green St., Yankton, SD 57078. Phone (605) 665-2833.

Computer-Controlled Alarm

VIN Systems (U.S.A.) announces the development of a complete intruder alarm system. The system features the Model V100 alarm interface, which plugs into either the VIC 20 or Commodore 64's user I/O port. VIN Systems (U.S.A.) furnishes software on either cassette or disk to operate the V100 alarm system.

The V100 alarm interface monitors four magnetic switches for an intruder. When an intruder is detected, a piercing 95db siren is activated. The system can be easily expanded to include pressure mat switches, glass breakage and vibration detectors, panic switches and much more.

The V100 system includes the interface, four magnetic switches, a 95db siren, 100 feet of wire and necessary software. The price of this total package is $69.00.



Cursed Your Computer Lately?

Newsletter Offers $25 To Print Your Troubles

Make a mistake with a hammer and you smash your thumb. (And perhaps call the hammer a few choice names.) Make a mistake with a computer and what happens? Chances are you have something worse than a smashed thumb and all the blue language you know won't begin to make you feel any better.

A new newsletter—titled "@*&!!%# COMPUTERS!" but pronounced "Expletive Computers"—might help you to ease the pain. Write down your tale of woe, complete with all the anxiety-producing details, and submit it for publication. If your true story is interesting, heartrending and/or amusing enough, it will be printed in an issue of "@*&!!%# COMPUTERS!", and you'll receive $25 for sharing your experience with the newsletter's readers.

Don't overlook any example that you or an acquaintance might offer—the repeated computer-generated invoices you've received for a bill that wasn't yours; the expensive equipment a friend purchased that now sits in a closet; the unbelievable hassles you went through to reconstruct a wiped-out memory.

Write it down! Then send it, with your name and address, to "Expletive Computers!", P.O. Box 553, Mt. Freedom, NJ 07970. If it appears in a future issue, you'll receive $25 for taking the trouble to reveal your trouble.

One-year subscriptions (ten issues) to "@*&!!%# COMPUTERS!" can be obtained for $6, also to be sent to the above address. The premier issue is scheduled for April, 1984. Sorry, "@*&!!%# COMPUTERS!" will *not* accept advertising, except from abacus manufacturers who've been in business 200 years or more....

New Software for Kids

Futurehouse, Inc. has announced the release of Playground Software, an educational series designed to be used with their Edumate Light Pen. *Alphabet Construction Set* and *Bedtime Stories* incorporate S.A.M™ voice synthesis, which has been licensed from Don't Ask Software, Inc. *Computer Crayons* and *Animal Crackers* focus on the graphics capabilities of the Edumate Light Pen.

Playground Software is targeted at children two to eight years old and is available on disk for the Commodore 64. The retail price is $29.95.

Resource for Washington, D.C., Computer Owners

Despite all the resources in the Washington, D.C. area for home computer owners, many area residents don't know where to go or have the time to discover what's available.

With the new year comes a unique publication that puts relevant information at their fingertips in a format that's quick to read and easy to understand. Specially designed for the Washington area home computer owner, *Home Computer Briefs: An Easy Guide to Information for Personal Computer Owners* is a monthly newsletter that guides owners to resources that will help them get the most out of their equipment. It features articles on training, repairs and other services; a word processing column; a calendar of users' group events (beginning February); reviews of microcomputer books and where they can be obtained locally; and a column through which readers can share good and bad experiences with their computers, salespeople, mail order houses and repair shops.

"We plan to spend time discovering everything available in this area to meet the needs of local home computer owners," says co-publisher Donna M. Hill.

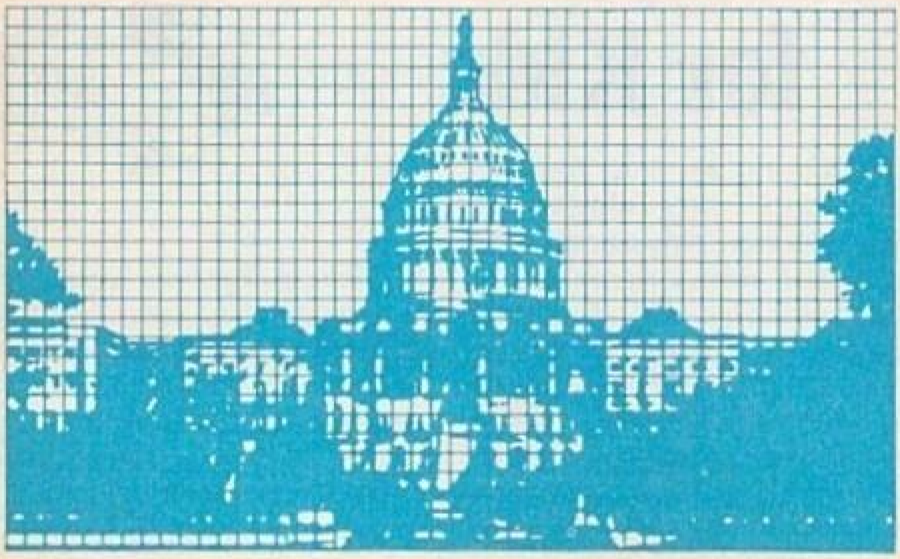
"Such an information source is really needed," she continues, "because with such a complex subject, and such a rapidly changing industry, it's impossible to find one person who knows everything about every computer, or even all the answers about any one, and that includes salespeople, computer buffs and the like."

Co-publishers Donna M. Hill and Jenner Mandel have backgrounds in the editorial and technical writing fields. At the time *Home Computer Briefs* was conceived, Hill and Mandel were both technical writers in the Division of Data Processing at the Federal Reserve Board. They are supported by an editorial board of computer analysts and home computer buffs.

Backed by a study confirming that one out of four home computer owners doesn't use his/her equipment, Hill and Mandel created *Home Computer Briefs* to fill the needs of these owners.

"We want to reach those who dropped out," says Hill, "cut time for those who are busy with other activities, help them all come closer to tapping the full potential of their equipment."

Anyone interested in a one-year subscription ($18) should call or write to Home Computer Briefs, 3421 M. St., N.W., Suite 1739, Washington, D.C. 20007, (202) 965-4428. **C**



The Commodore Puzzle Book

Reviewed by Lou Hinshaw

Authors: Nevin B. Scrimshaw and Gordon Lee
Publisher: Birkhauser Boston, Inc. 380 Green Street, Cambridge, MA 02139
Date: 1983
Price: $7.95
Pages: 125 pages, softcover

Generally, programming skills grow from plateau to plateau. That is to say, a fair writer of programs might slog along for quite some time doing fairly good work and not improving any to speak of. But one fine day, the fairly good skill, our intrepid programmer finds, is not good enough. He must grow or fail and for a few days at the very best, his life is a life of anguish as he sweats out an approach to a very sticky problem. Often he is not the only one sweating. A sticky problem can be the end of a job in some cases, competition being what it is in many shops.

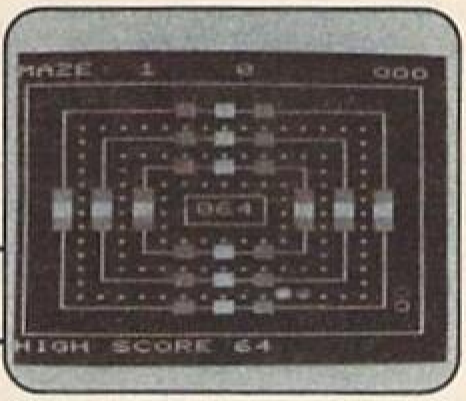
The foresighted programmer, then, will welcome *The Commodore Puzzle Book*. Here, for use with a home computer, are a host of brain sharpeners, a veritable landslide of preparation for the next hard one at the shop, and *fun* to boot!

Ostensibly written for Commodore computers and BASIC, the problems offered here will be a challenge on any machine and in any language I know. For fun and profit, I recommend *The Commodore Puzzle Book*. The most-used remark in the application of the book is going to be, "This problem is not as blamed simple as I thought!"

The book is in three parts: puzzles, programs, and solutions. For the lazy computerist, let me warn you, keep out! The programs in the middle often only suggest approaches and the solutions are almost limited to checksums. The problems, fifty in all, are word pictures with logical difficulty and simple math. The subjects range from religion through carpentry to cryptology. (Is that a full circle?) A sample program is provided for each problem, and the programs can be run on most home computers.

The Commodore Puzzle Book: BASIC Brainteasers, by Lee and Scrimshaw, is about as good a way to spend eight dollars as I can think of. "Kids of all ages" may love or hate it but few, I think, will ignore this book. A great gift. **C**

Congratulations Contest Winner



David Liu, Diamond Bar, California, for *Crash*

Crash *For the Unexpanded VIC 20*

by David Liu

Try to drive your car around the track collecting dots without *Crash*-ing into the rival car that's out to get you. Move from outside tracks to inside tracks using the "gates" located between checkered areas. You control your car from the keyboard, using F5 for up, F7 for down, S for left and D for right. Gentlemen (and women), start your engines!

```
0 POKE 36879,9:POKE 36878,15
  :PRINT"[CLEAR,WHITE,DOWN]";
  :GOSUB 700:MO=4
1 TH=0:MO=MO-1:IF MO<0 THEN MO=2
2 A=-1:B=22:C=1:D=-22:S=7756:Z=0
3 K=-1:L=-22:M=1:N=22:R=8117
10 PO=PO+1:C1=0:PRINT"[CLEAR,
   DOWN2,WHITE,CMDR A,SHFT *19,
   CMDR S]"
11 PRINT"[SHFT -]...............
   ....[SHFT -]"
12 PRINT"[SHFT -].[CMDR A,
   SHFT *5,CMDR +,SHFT *,RVS]
   [RVOFF,SHFT *,CMDR +,SHFT *5,
   CMDR S].[SHFT -]"
13 PRINT"[SHFT -].[SHFT -]
   ...............[SHFT -].
   [SHFT -]"
14 PRINT"[SHFT -].[SHFT -].
   [CMDR A,SHFT *3,CMDR +,SHFT *,
   RVS] [RVOFF,SHFT *,CMDR +,
   SHFT *3,CMDR S].[SHFT -].
   [SHFT -]"
15 PRINT"[SHFT -].[SHFT -].
   [SHFT -]...........[SHFT -].
   [SHFT -].[SHFT -]"
16 PRINT"[SHFT -].[SHFT -].
   [SHFT -].[CMDR A,SHFT *,
   CMDR +,SHFT *,RVS] [RVOFF,
   SHFT *,CMDR +,SHFT *,CMDR S].
```

(Continued On Next Page)

The Commodore Challenge

Prizes • Prizes • Prizes

Thanks to all of you who submitted programs. They were all, as usual, great fun. (We never seem to have any problems getting volunteer judges for this department, for some reason.) And keep up the good work. Maybe next time the winner will be you.

If you've been playing around at home developing original games and programs for your unexpanded VIC 20, send your best—on cassette or disk, please—to the Commodore Challenge contest. Include a brief description of the program's purpose, including documentation on how to use it. If it's a game, be sure to include instructions.

Programs requiring memory expansion are eligible, too, but will not be published unless space allows.

Winners will receive a VIC 20 8K Memory Expander Cartridge. All entries become the property of Commodore Business Machines, Inc., upon submission.

Fill out the entry form below, and submit it with your game or program to:

**Commodore Business Machines, Inc.
1200 Wilson Drive, West Chester, PA 19380
Attn: POWER/PLAY**

Commodore Challenge Contest... Entry Form

Name__________ Age______ Phone______

Address__________ Program Title______

City__________ State______ Zip______

I understand that my software entry becomes the property of Commodore Business Machines, Inc., upon submission.

Signature__________

Parent's signature, if contestant is minor__________

Void Where Prohibited

```
    [SHFT -].[SHFT -].[SHFT -]"
 17 PRINT"[SHFT -].[SHFT -].
    [SHFT -].[SHFT -].......
    [SHFT -].[SHFT -].[SHFT -].
    [SHFT -]"
 18 PRINT"[SHFT -].[CMDR +].
    [CMDR +].[CMDR +].[CMDR A,
    SHFT *3,CMDR S].[CMDR +].
    [CMDR +].[CMDR +].[SHFT -]"
 19 PRINT"[SHFT -].[RVS] [RVOFF].
    [RVS] [RVOFF].[RVS] [RVOFF].
    [SHFT -,SPACE3,SHFT -].[RVS]
    [RVOFF].[RVS] [RVOFF].[RVS]
    [RVOFF].[SHFT -]"
 20 PRINT"[SHFT -].[CMDR +].
    [CMDR +].[CMDR +].[CMDR Z,
    SHFT *3,CMDR X].[CMDR +].
    [CMDR +].[CMDR +].[SHFT -]"
 21 PRINT"[SHFT -].[SHFT -].
    [SHFT -].[SHFT -].......
    [SHFT -].[SHFT -].[SHFT -].
    [SHFT -]"
 22 PRINT"[SHFT -].[SHFT -].
    [SHFT -].[CMDR Z,SHFT *,
    CMDR +,SHFT *,RVS] [RVOFF,
    SHFT *,CMDR +,SHFT *,CMDR X].
    [SHFT -].[SHFT -].[SHFT -]"
 23 PRINT"[SHFT -].[SHFT -].
    [SHFT -]...........[SHFT -].
    [SHFT -].[SHFT -]"
 24 PRINT"[SHFT -].[SHFT -].
    [CMDR Z,SHFT *3,CMDR +,SHFT *,
    RVS] [RVOFF,SHFT *,CMDR +,
    SHFT *3,CMDR X].[SHFT -].
    [SHFT -]
 25 PRINT"[SHFT -].[SHFT -]
    ...............[SHFT -].
    [SHFT -]":PRINT"[SHFT -].
    [CMDR Z,SHFT *5,CMDR +,SHFT *,
    RVS] [RVOFF,SHFT *,CMDR +,
    SHFT *5,CMDR X].[SHFT -]"
 26 PRINT"[SHFT -]...............
    ....[SHFT -]":PRINT"[CMDR Z,
    SHFT *19,CMDR X]"
 27 T=PEEK(S):CH=CH+1
    :IF CH>1 THEN GOSUB 600
 28 PRINT"[HOME,DOWN21]
    HIGH SCORE";HI:Z=Z+1
    :IF Z<2 THEN 41
 29 PRINT"[HOME,DOWN]MAZE
    :";PO;TAB(10);TH;
    :FOR I=7720 TO 7723:POKE I,32
    :NEXT:FOR I=7720 TO 7720+2-ED
 30 POKE I,87:NEXT:Z=0
    :IF PEEK(S+A)=46 THEN 33
```

```
 31 IF PEEK(S+A)=87 THEN 500
 32 IF PEEK(S+A)<>32 THEN E=A:A=B
    :B=C:C=D:D=E
 33 IF PEEK(S-22)=160 THEN A1=-44
 34 IF PEEK(S+22)=160 THEN A1=44
 35 IF PEEK(S-1)=160 THEN A1=-2
 36 IF PEEK(S+22)=160 AND
    PEEK(S-22)=160 THEN
    A1=44*(INT(RND(1)*-3)+2)
 37 IF PEEK(S+1)=160 THEN A1=2
 38 IF PEEK(S+1)=160 AND
    PEEK(S-1)=160 THEN
    A1=2*(INT(RND(1)*-3)+2)
 39 IF ABS(A1)=3 OR ABS(A1)=66
    THEN A1=0
 40 POKE S,T:S=S+A+A1:T=PEEK(S)
    :POKE S,81:A1=0:IF T=87 THEN
    500
 41 X=0:GET A$:IF A$=""THEN 45
 42 IF A$="[F5]"AND ABS(K)=1 THEN
    X=-44:V=-22
 43 IF A$="[F7]"AND ABS(K)=1 THEN
    X=44:V=22
 44 IF A$="S"AND ABS(K)=22 THEN
    X=-2:V=-1
 45 IF A$="D"AND ABS(K)=22 THEN
    X=2:V=1
 46 IF PEEK(R+V)<>102 THEN X=0
 47 SO=SO+2:POKE R,32
    :IF PEEK(R+K+X)=46 THEN
    SC=SC+PO:POKE 36876,180
    :C1=C1+1:GOTO 51
 48 IF X<>0 THEN 51
 49 IF PEEK(R+K)=81 THEN 500
 50 IF PEEK(R+K)<>32 THEN P=K:K=L
    :L=M:M=N:N=P:SO=0
 51 S2$=STR$(SC):R=R+K+X:POKE R,87
    :SC$="000"+RIGHT$(S2$,
    LEN(S2$)-1)
 53 PRINT"[HOME,DOWN11,RIGHT9]";
    RIGHT$(SC$,3)
 55 X=0:IF C1=160 THEN 1
 56 IF SC=999 THEN TH=TH+1:SC=0
 57 IF SC+(TH*1000)>HI THEN
    HI=SC+(TH*1000)
 60 POKE 36876,0:GOTO 28
 500 POKE R,86:SO=0
 503 FOR I=230 TO 128 STEP-5
     :FOR O=1 TO 3:POKE 36873+O,I
     :NEXT O,I
 504 FOR I=1 TO 4:POKE 36873+I,0
     :NEXT:LS$=SC$
 505 ED=ED+1:IF ED=3 THEN ED=0
     :FOR I=1 TO 4000:NEXT
     :PRINT"[CLEAR]":GOTO 0
```

```
507 POKE R,32:POKE S,T
    :IF T=87 THEN POKE S,32
508 A=-1:B=22:C=1:D=-22:S=7756
    :Z=0:K=-1:L=-22:M=1:N=22
    :R=8117
509 T=PEEK(S):FOR I=1 TO 500:NEXT
    :GOTO 28
600 POKE 36877,0:FOR I=7724 TO
    8142
601 IF PEEK(I)=102 THEN POKE I,
    160:POKE 36876,222
    :FOR O=1 TO 10:NEXT:GOTO 603
602 IF PEEK(I)=160 THEN POKE I,
    102:POKE 36876,250
    :FOR O=1 TO 10:NEXT
603 POKE 36876,0:NEXT:RETURN
700 E$="[CLEAR,DOWN]
     GET READY TO PLAY...[SPACE2]
    ":FOR I=1 TO 23
    :PRINT MID$(E$,I,1);:CH=0
701 IF MID$(E$,I,1)<>" "THEN
    POKE 36875,235
702 FOR O=1 TO 99:NEXT
    :POKE 36875,0:NEXT
    :B$(1)="[RIGHT,DOWN] [RED,
    SHFT N,CMDR +2,DOWN,LEFT3,
    CMDR +,DOWN,LEFT,SHFT M,
    CMDR +2,RIGHT]":Z=0:SC=0:C1=0
    :PO=0
703 B$(2)="[WHITE,CMDR +2,SHFT M,
    DOWN,LEFT3,CMDR +,SHFT L,
    CMDR +,DOWN,LEFT3,CMDR +,
    SHFT M2,RIGHT]"
704 B$(3)="[BLUE,SHFT N,CMDR +,
    SHFT M,DOWN,LEFT3,CMDR +,
    CMDR POUND,CMDR +,DOWN,LEFT3,
    CMDR +] [CMDR +,RIGHT]"
705 B$(4)=" [WHITE,SHFT N,
    CMDR +2,DOWN,LEFT3,SHFT M2,
    DOWN,LEFT3,CMDR +2,SHFT N,
    RIGHT]"
706 B$(5)="[RED,CMDR +] [CMDR +,
    DOWN,LEFT3,CMDR +3,DOWN,
    LEFT3,CMDR +] [CMDR +,DOWN,
    RIGHT5]"
707 C$="[GREEN]BY DAVID LIU
    [WHITE]":D$="TO PLAY... HIT
    [RVS]RETURN[RVOFF]"
708 SC$=STR$(VAL(RIGHT$(SC$,
    3))+TH*1000):IF SC$=""THEN
    SC$="000"
709 C1$="LAST SCORE:"+SC$
    :FOR I=1 TO 10
710 POKE 36876,150+(I*5)
    :POKE 8054+I,81
    :POKE 8075-I,87
    :FOR O=1 TO 100:NEXT
    :POKE 8054+I,32
711 POKE 8075-I,32:NEXT:R=8054+I
    :POKE R,86:POKE R-1,81
    :POKE 36876,0
715 FOR I=1 TO 5:PRINT B$(I);
    :NEXT:POKE 36877,200
    :FOR I=15 TO 0 STEP-.05
    :POKE 36878,I:NEXT
720 POKE 36878,15:POKE 36877,0
731 FOR I=1 TO LEN(C$)
    :PRINT MID$(C$,I,1);
    :IF MID$(C$,I,1)<>" "THEN
    POKE 36875,235
733 FOR O=1 TO 100:NEXT
    :POKE 36875,0:NEXT
    :PRINT"[DOWN2]
735 FOR I=1 TO LEN(C1$)
    :PRINT MID$(C1$,I,1);
    :IF MID$(C1$,I,
    1)<>" "THEN POKE 36875,235
736 FOR O=1 TO 100:NEXT
    :POKE 36875,0:NEXT
    :PRINT"[DOWN2]
737 FOR I=1 TO LEN(D$)
    :PRINT"[UP]";LEFT$(D$,I)
    :IF MID$(D$,I,1)<>" "THEN
    POKE 36875,235
738 FOR O=1 TO 100:NEXT
    :POKE 36875,0:NEXT
740 FOR I=1 TO 650:GET A$
    :IF A$=""THEN 750
745 IF ASC(A$)=13 THEN RETURN
750 NEXT:PRINT"[CLEAR,DOWN,RVS,
    SPACE4]THE CONTROLLS
    :[SPACE4]"
755 PRINT"[SPACE2,CMDR @,DOWN,
    LEFT2,CMDR M]S[CMDR G,DOWN,
    LEFT2,CMDR T,RIGHT,UP]
    :LEFT[DOWN]"
756 PRINT"[RIGHT2,CMDR @,DOWN,
    LEFT2,CMDR M]D[CMDR G,DOWN,
    LEFT2,CMDR T,RIGHT,UP]
    :RIGHT[DOWN]"
757 PRINT"[RIGHT,CMDR @2,DOWN,
    LEFT3,CMDR M]F5[CMDR G,DOWN,
    LEFT3,CMDR T2,RIGHT,UP]
    :UP[DOWN]"
758 PRINT"[RIGHT,CMDR @2,DOWN,
    LEFT3,CMDR M]F7[CMDR G,DOWN,
    LEFT3,CMDR T2,RIGHT,UP]:DOWN"
```

C

the joystick lunatic

The Fantastic Computer Olympics

by Steve Finkel

Just when you thought it was safe to go back into the computer game store.... comes the Joystick Lunatic (in 3-D, if you wear those special paper glasses). The Joystick Lunatic is a new feature in *Power/Play* that will focus its attention on the world of computer games.

The computer gamer and computer game genre are an unusual blend of human persistence, technical equipment, playing techniques, etc., with game styles and forms ranging from the obscure to the obvious, from the apocalyptic to the whimsical. They afford us a chance to observe, which often leads to insight, understanding and amusement. The computer gaming arena readily lends itself to analysis. Frequently, the computer gamers themselves enter analysis. (That was a cheap one-liner, so ignore it. The Joystick Lunatic abhors cheap one-liners, preferring instead the lengthier cheap two-liner.)

Never in this column will you find a joke that is intended to get a cheap laugh at someone else's expense, such as this: How many game players does it take to change a lightbulb? This is not what anyone would consider an insightful observation, and really just doesn't make the high score page on my monitor. No, what the Joystick Lunatic is more concerned with is thoughtful, meaningful information that will expand the reader's scope and possibly change the way he or she looks at the world. Then again, perhaps the aim of this column is just to fill two pages. In either case, that brings us to the topic at hand.

With all the excitement over the upcoming summer Olympics, it might surprise people to learn that every year for the last four years there has been an Olympic-style competition for computer gamers from all over the world. The events and contestants have become more sophisticated in the last few years, compared to when the games began. The one thing that has always remained constant is the spirit that pervades the games.

In the competition between gamers of all nations, the competitors all share one quality: they are rotten losers.

Thus, under the banner of the games (the five interlocking monitor screens) voices of the competitors chime together in a universal message unhindered by language differences: "CHEATER!!"

Bad sportsmanship aside, the games do represent computer game competition at its very finest. Some of the best game people in the world compete, the only restrictions being that they must be amateurs (the Communist Bloc nations are annually accused of sending professional gaming teams) and that they don't take anabolic steroids to enhance their game-playing prowess. Some of the events that have become standard in the competition are:

Joystick Toss—Contestants, after a very poor round of their favorite games in which they lose all their men because the joystick is broken, must hurl their joysticks as far as they can. Those throwing the joystick the greatest distance are judged winners. In case of a tie, the gold medal is awarded to the contestant whose joystick suffers the most damage because of the toss.

World record holder in this event is Sergei Popanov, a Russian competitor who threw his joystick with such force that he not only set a distance record, but knocked a local government official out cold when the joystick struck him on the forehead. Sergei is not eligible for parole until early March, 2041, but plans to continue his promising career while in custody by hurling invisible joysticks. His constant imaginary practice sessions will

Steve Finkel

serve a secondary purpose: a serious attempt to substantiate his lawyer's insanity defense plea. Best of luck, Sergei.

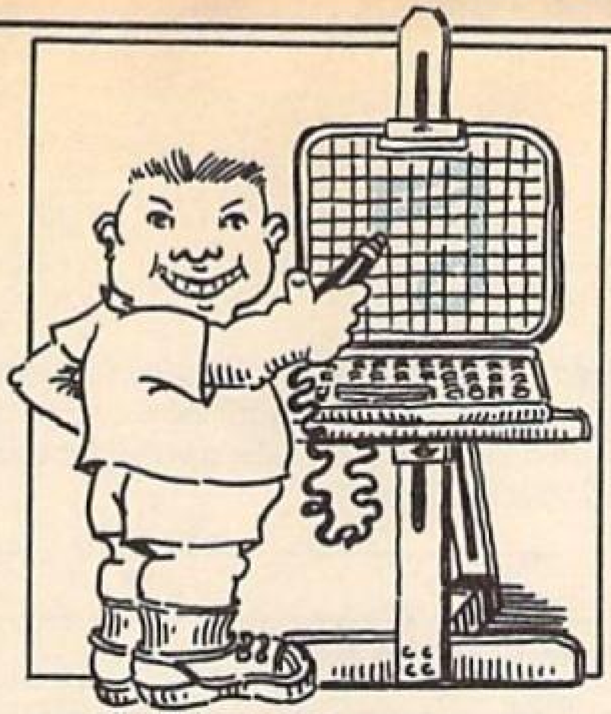
Speed Resetting—All world-class game players have to be adept at hitting the restore key when they experience extreme failure in the early rounds of a game that began as an attempt at a new high score. The more experienced the gamer, the faster the restore button is pressed and the game reset. The world-class resetters often seem to anticipate the loss of a ship so readily that if you blink, you might not even know they re-started the game.

From its inception, this event has been dominated by the team from Ireland. (That's right, the Irish Resetters. Sorry.) Ending their dynasty, however, was a young Yugoslavian, Feodor Dubrovnik, whose resetting exploits are legendary. Feodor, in point of fact, never really understood what was going on, but this only increased his resetting prowess.

Legend has it that Feodor, in his prime, never actually finished a computer game, instead pressing reset an average of 68 times a minute. When asked if possibly it might be more fun to actually play the game, young Dubrovnik replied "What? You mean there's more?". It seems that Feodor was under the impression that computer games merely entailed the pressing of the reset key when the title screen came up, and he really didn't see what all the fuss was about. Upon learning otherwise, Dubrovnik left the competitive reset arena and has now devoted himself to trying to popularize the up-and-coming new sport of Jello surfing.

In the competition between gamers of all nations, the competitors all share one quality: they are rotten losers.

Freestyle Lightpen Maneuvering—Participants manipulate lightpens over a monitor screen in a dramatic style to a musical piece. Points are given for originality, style,



overall concept, and appropriateness of musical selection. Gershwin's Rhapsody in Blue and Strauss' Blue Danube Waltz are traditional favorites for background music.

One of the major surprises in the Summer games was a silver medal being awarded to a punk rock contestant, 6'5" 240-pound Lyle "Filthy" Axemurderer, whose self-composed music served as a major boost to the medal. His song, "If I Don't Win a Medal, I'll Set the Judges' Heads on Fire," apparently was received with overwhelming approval by the judges, who gave him an excellent score. Other contestants were less than impressed by what they termed his "artless scribbling" with the lightpen, but the judges' decision was upheld by unanimous acclaim when he unveiled, in a special exhibition, another composition, "I Know Where You All Live and What Your Children Look Like".

Between Rounds Snack Relay—The purpose of this event is to dash to the refrigerator and fix a substantial snack, return to the game and consume said snack, eating as the game continues by controlling the joystick with one hand and eating with the other.

Lloyd Glutton, of England, once consumed a nine-course meal featuring hot soup, shrimp cocktails and three different entrees, without losing a man. However, this great performance did not make the record books. At the weigh-ins Lloyd qualified as a middleweight, but after his fine effort, his classification was a heavyweight. Not only was Mr. Glutton's performance overlooked in the record books because of this, but his wife refused to let him reenter the house until he shed his excess poundage.

In last year's competition, a near tragedy occurred when the French entrant Bridget Fromage lost her concentration and inadvertantly manipulated a croissant with her right hand and used her left to swallow her joystick. She was rushed to the hospital where they were able to remove only part of the implement. She has nearly completely recovered, except that now her belly button functions as a fire button. If she gets poked in the stomach, she automatically expels whatever she has in her mouth. This is often embarrassing at finer restaurants, but sometimes there is a price to pay for greatness.

These are only some of the popular and entertaining events that highlight the Games Olympics. It may not match the real Olympics in excitement, drama or even sponsors, but it does have some legitimate notoriety. It seems that the president of the International Computer Games Olympics Committee somehow misled the 1981 Luxembourg Gaming Team, causing them to believe that the games were to be held at the South Shetland Islands (just off the Antarctic Peninsula). Dedicated gamers as the Luxembourgers are, the entire team trekked through the frozen ice floes, only to find that the games were being held elsewhere. Although they missed out on that year's competition, the devoted Luxembourg team did manage to make the early editions of the world's newspapers and bring glory to the good name of Luxembourg and gamers in general, being the first group of people ever to place a joystick on the South Pole.

As I said before, sometimes there is a price to pay for greatness. The Luxembourg squad was not properly prepared to face the bitter climate, making no special arrangements for cold-weather footwear. After wandering around the Antarctic for six weeks wearing sneakers, each team member suffered from acute frostbite on their toes, often referred to in the sporting world as the agony of de feet. (Was that you I heard groaning just now?)

By the way, the answer to the lightbulb question is none, since all gamers know that the monitor screen looks better in the dark, anyhow. **C**

high scores

In order to have your high score published, you must send in a photo of the winning screen showing the score. Make sure your name and address are on the back of the photo. If your score is high, we'll run your name and score as soon as we can get it in. Be patient. We work far ahead putting an issue together, so it might take a while for your score to show up.

VIC 20 Games		**Score**
BLUE MEANIES	Peter J. Cwik, Detroit, MI	1,440
CAR CHASE	Alan Howard, Silver Lake, KS	90,440
CLOWNS	Jon Kirkbride, Cedars, PA	97,780
COSMIC CRUNCHER	Joe Lindo, Burlington, Ontario	998,180
DRAW POKER	R. Callia, Torrance, CA	17,410
GORF	Thad Kizior, Ft. Murray, Alberta	354,220
JUPITER LANDER	Christopher Champlain, St. Petersburg, FL	207,400
MIDNIGHT DRIVE	Nathan Mehl, Newark, DE	14.11 km
MOLE ATTACK	Ryan Phillips, Montpelier, VT	427
MONEY WARS	Bob Grant, Bensalem, PA	69,140
OMEGA RACE	Jim Nelson, S. Elgin, IL	1,502,300 (5 ships)
PINBALL	Joe Ferrari, Commodore, Toronto	1,500,000
RADAR RAT RACE	Jennifer Zaruk, San Diego, CA	147,240
RAID ON FORT KNOX	Chad Strait, Ankeny, IA	8,414
SEA WOLF	Alieia Kondalski, Toledo, OH	29,500
SKY IS FALLING	Christina Zip, Toledo, OH	22,080
SLITHER	Richard R. Sunny, Sr., Catasauqua, PA	293
SUPER ALIEN	Robert Schaeffer, Brookline, MA	45,700
SUPER SLITHER	Norbert Scheel, Mississauga, Ontario	203
SUPER SLOT	Judith Wright, Cheektowaga, NY	10,960 coins
VIC AVENGER	Nathan Brown, Newark, DE	23,120

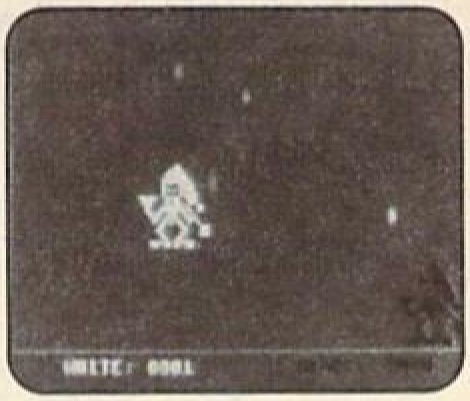
Commodore 64 Games		
AVENGER	Britt Feldhaus, Edina, MN	33,600
CLOWNS	Carla Heiland, S. Pasadena, CA	94,180
KICKMAN	Alan Miller, Cincinnati, OH	530,208
LEMANS	Chris Waldroff, Deephaven, MN	274,460
OMEGA RACE	Alan Kalis, Athen, OH	600,500 (4 ships)
STARPOST	Carl Peterson, Poway, CA	329,900

If your score didn't set a record this time, keep playing! Maybe you'll topple these champion gamesters next time! **C**

Skycatcher

by Len Lindsay

This program is written for the Commodore 64 using the COMAL language and disk drive. COMAL is a structured programming language available on disk for the Commodore 64 for $19.95 from the COMAL Users Group, U.S.A., 5501 Groveland Terrace, Madison, Wisconsin, 53716. It is also available on disk for PET/CBM for $14.95. For more information on COMAL, contact Len Lindsay at the address above.



Skycatcher

```
0010 //DELETE "0:SKY'CATCHER21"
0020 // COMAL PROGRAM: SKY CATCHER
0030 //SAVE   "0:SKY'CATCHER23"
0040 // BY: CAPTAIN COMAL & FRIENDS
0050 // FRIENDS: LEN LINDSAY, MOENS
     KJAER, LARS LAURSEN, JENS ERIK
     JENSEN
0060 // PLACED INTO PUBLIC DOMAIN
0070 //
0080 DIM MOBS$ OF 64, CH$ OF 1, SCORE$
     OF 4
0090 PAGE
0100 PASS "I0" //INIT DISK
0110 READ'SPRITES
0120 BACKGROUND 2
0130 BORDER 2
0140 PENCOLOR 1
0150 INSTRUCTIONS
0160 SET'DIFFICULTY
0170 INIT
0180 DONE:=FALSE
0190 REPEAT
0200  // DELTA1 & DELTA2 ARE AMOUNT TO
     MOVE SIDEWAYS
0210  X1:+DELTA1; X2:+DELTA2
0220  // CHECK NOT PAST BORDER
0230  IF X1<1 THEN X1:=1; DELTA1:=0
0240  IF X2<1 THEN X2:=1; DELTA2:=0
0250  IF X1>270 THEN X1:=270; DELTA1:=0
0260  IF X2>270 THEN X2:=270; DELTA2:=0
0270  // MOVE PLAYER1
0280  IF DELTA1>=0 THEN
0290   SPRITEPOS 0,X1,Y1
0300   SPRITEPOS 1,X1,Y1
0310  ELSE
0320   SPRITEPOS 1,X1,Y1
0330   SPRITEPOS 0,X1,Y1
0340  ENDIF
0350  // MOVE PLAYER2
0360  IF DELTA2>=0 THEN
0370   SPRITEPOS 2,X2,Y2
0380   SPRITEPOS 3,X2,Y2
0390  ELSE
0400   SPRITEPOS 3,X2,Y2
0410   SPRITEPOS 2,X2,Y2
0420  ENDIF
0430  GET'NEXT'MOVE
0440  IF SPRITECOLLISION(1,TRUE) OR
      SPRITECOLLISION(3,FALSE) THEN
      COLLISION
0450  FALLING'PIECES
0460 UNTIL DONE
0470 SETTEXT
0480 //
0490 PROC INITGAME
0500  DIM PIECES(NUM'PIECES,2)
0510  NEW'PIECES
0520  FOR I:=4 TO NUM'PIECES+3 DO
```

(Continued On Next Page)

```
0530   SPRITECOLOR I,8
0540   IDENTIFY I,2
0550   SPRITESIZE I,0,0
0560  ENDFOR I
0570  SPRITECOLOR 5,7
0580  PENCOLOR 8
0590  MOVE'PIECES
0600  DRAW'BORDER
0610  BLACK:=0; WHITE:=0
0620  POINT
0630  I:=SPRITECOLLISION(1,TRUE)
0640 ENDPROC INITGAME
0650 //
0660 PROC FALLING'PIECES
0670  FALLING
0680  MOVE'PIECES
0690 ENDPROC FALLING'PIECES
0700 //
0710 PROC MOVE'PIECES
0720  FOR I:=1 TO NUM'PIECES DO
      SPRITEPOS I+3,PIECES(I,1),
      PIECES(I,2)
0730 ENDPROC MOVE'PIECES
0740 //
0750 PROC FALLING
0760  FOR I:=1 TO NUM'PIECES DO
0770   PIECES(I,2):-I*FALL'RATE
0780   IF PIECES(I,2)<38 THEN PIECES
       (I,2):=200; PIECES(I,1):=RND
       (1,270)
0790  ENDFOR I
0800 ENDPROC FALLING
0810 //
0820 PROC COLLISION
0830  WHITE:+SPRITECOLLISION(1,FALSE)
0840  BLACK:+SPRITECOLLISION(3,FALSE)
0850  TOP'OR'BOTTOM
0860  FOR I:=1 TO NUM'PIECES DO
0870   IF SPRITECOLLISION(I+3,FALSE)
       THEN
0880    PIECES(I,1):=RND(1,270)
0890    PIECES(I,2):=200
0900   ENDIF
0910  ENDFOR I
0920  POINT
0930  MOVE'PIECES
0940  I:=SPRITECOLLISION(1,TRUE)
0950 ENDPROC COLLISION
0960 //
0970 PROC POINT
0980  PENCOLOR 1
0990  STR(WHITE,SCORE$)
1000  PLOTTEXT 40,8,"WHITE: "+SCORE$
1010  PENCOLOR 0
1020  STR(BLACK,SCORE$)
1030  PLOTTEXT 200,8,"BLACK:  "+SCORE$
1040 ENDPROC POINT
1050 //
1060 PROC STR(I,REF SCORE$) CLOSED
1070  SCORE$:="    "; J:=1000
1080  FOR K:=1 TO 4 DO
1090   SCORE$(K):=CHR$(48+I DIV J)
1100   I:=I MOD J; J:=J/10
1110  ENDFOR K
1120 ENDPROC STR
1130 //
1140 PROC NEW'PIECES
1150  FOR I:=1 TO NUM'PIECES DO
1160   PIECES(I,1):=RND(8,280)
1170   PIECES(I,2):=RND(10,80)
1180  ENDFOR I
1190 ENDPROC NEW'PIECES
1200 //
1210 PROC TOP'OR'BOTTOM
1220  // PLAYER WITH MOST POINTS IS ON
      TOP
1230  IF WHITE<=BLACK THEN
1240   Y1:=BOTTOM; Y2:=TOP
1250  ELSE
1260   Y1:=TOP; Y2:=BOTTOM
1270  ENDIF
1280 ENDPROC TOP'OR'BOTTOM
1290 //
1300 PROC INSTRUCTIONS
1310  PAGE
1320  PRINT "THE FIGURES ARE MOVED AS
      FOLLOWS:"
1330  PRINT
1340  PRINT
1350  PRINT "  THE WHITE PLAYER:"
1360  PRINT
1370  PRINT "A   MOVE TO THE LEFT"
1380  PRINT "S   STOP MOVING"
1390  PRINT "D   MOVE TO THE RIGHT"
1400  PRINT
1410  PRINT "  THE BLACK PLAYER:"
1420  PRINT
1430  PRINT ":   MOVE TO THE LEFT"
1440  PRINT ";   STOP MOVING"
1450  PRINT "=   MOVE TO THE RIGHT"
1460  PRINT
1470  PRINT "Q   QUIT"
1480  PRINT
1490  PRINT "TRY TO CATCH PIECES OF
      FALLING SKY!"
1500  PRINT "CATCH THEM ON THE PLATE
      YOUR ELF HOLDS"
1510  PRINT
1520  PRINT " START BY ANY KEY:"
1530  REPEAT
```

```
1540  UNTIL KEY$<>CHR$(0)
1550 ENDPROC INSTRUCTIONS
1560 //
1570 PROC INIT
1580  SETGRAPHIC 0
1590  HIDETURTLE
1600  INITGAME
1610  SPRITECOLOR 0,1
1620  SPRITECOLOR 1,1
1630  SPRITECOLOR 2,0
1640  SPRITECOLOR 3,0
1650  SPRITESIZE 0,1,1
1660  SPRITESIZE 1,1,1
1670  SPRITESIZE 2,1,1
1680  SPRITESIZE 3,1,1
1690  IDENTIFY 0,0
1700  IDENTIFY 2,0
1710  IDENTIFY 1,1
1720  IDENTIFY 3,1
1730  D:=7.5; DELTA1:=D; DELTA2:=D
1740  BOTTOM:=64; TOP:=114
1750  X1:=120; X2:=120
1760  Y1:=BOTTOM; Y2:=TOP
1770 ENDPROC INIT
1780 //
1790 PROC GET'NEXT'MOVE
1800  CH$:=KEY$
1810  IF CH$<>CHR$(0) THEN
1820   IF CH$="A" THEN DELTA1:=-D
1830   IF CH$="D" THEN DELTA1:=D
1840   IF CH$="S" THEN DELTA1:=0
1850   IF CH$=":" THEN DELTA2:=-D
1860   IF CH$="=" THEN DELTA2:=D
1870   IF CH$=";" THEN DELTA2:=0
1880   IF CH$=" " THEN TEMP:=Y1; Y1:
       =Y2; Y2:=TEMP
1890   IF CH$="Q" OR CH$="@" THEN DONE:
       =TRUE
1900  ENDIF
1910 ENDPROC GET'NEXT'MOVE
1920 //
1930 PROC DRAW'BORDER
1940  MOVETO 0,20
1950  DRAWTO 0,199
1960  DRAWTO 319,199
1970  DRAWTO 319,20
1980  DRAWTO 0,20
1990 ENDPROC DRAW'BORDER
2000 //
2010 PROC PAGE
2020  PRINT CHR$(147), //CLEAR SCREEN
2030 ENDPROC PAGE
2040 //
2050 PROC SET'DIFFICULTY
2060  PAGE
```

```
2070  REPEAT
2080   INPUT "HOW MANY FALLING PIECES
       (1-4): ": NUM'PIECES
2090  UNTIL NUM'PIECES>=1 AND
      NUM'PIECES<=4
2100  REPEAT
2110   INPUT "FALLING SPEED (SLOW 1-4
       FAST): ": FALL'RATE
2120  UNTIL FALL'RATE>=1 AND FALL'RATE
      <=4
2130 ENDPROC SET'DIFFICULTY
2140 //
2150 PROC READ'SPRITES
2160  PRINT "SETTING UP SPRITE IMAGES"
2170  DIM IMAGE$ OF 64
2180  FOR SHAPES:=0 TO 2 DO
2190   IMAGE$:="" //INIT
2200   FOR BYTES:=1 TO 64 DO
2210    READ ITEM
2220    IMAGE$:=IMAGE$+CHR$(ITEM)
2230   ENDFOR BYTES
2240   DEFINE SHAPES,IMAGE$
2250  ENDFOR SHAPES
2260 ELF'IMAGE:
2270  DATA 0,12,0,0,62,0,0,127,0,0,255,
      128,0,255,192,0
2280  DATA 131,224,0,171,224,0,131,240,
      0,124,240,120,108,112,48,254
2290  DATA 140,49,255,76,27,57,160,14,
      124,240,4,214,72,1,131,7
2300  DATA 3,1,135,1,131,7,0,198,0,15,
      239,224,14,108,224,0
2310 CATCH'PLATE:
2320  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0
2330  DATA 0,0,0,0,0,0,0,0,252,0,0,0,0,
      0,0,0
2340  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0
2350  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0
2360 SKY'PIECE:
2370  DATA 48,0,0,120,0,0,252,0,0,252,
      0,0,252,0,0,120
2380  DATA 0,0,48,0,0,0,0,0,0,0,0,0,0,
      0,0,0
2390  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0
2400  DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,
      0,0
2410 ENDPROC READ'SPRITES
2420 //
```

C

```
460 GOTO 120
500 LS=0:IF SIZE>=IS THEN
    IS=IS+10:POT=POT+1:GOTO 1000
520 PRINT"POT TOO BIG":POT=POT+1
    :CS=CS+1:IF CS>3 THEN
    D$="OVERPOTTING":GOTO 3000
530 GOTO 2000
600 IF POT>0 THEN POKE CH+456,4
    :POKE SC+456,160
    :POKE CH+478,4:POKE SC+478,
    160
615 IF POT>0 THEN POKE CH+455,4
    :POKE SC+455,95:POKE CH+457,4
    :POKE SC+457,105
620 IF POT>1 THEN POKE SC+455,160
    :POKE SC+457,160
    :POKE SC+477,160
    :POKE CH+477,4
625 IF POT>1 THEN POKE SC+479,160
    :POKE CH+479,4:POKE SC+454,95
    :POKE CH+454,4
626 IF POT>1 THEN POKE SC+458,105
    :POKE CH+458,4
630 IF POT>2 THEN POKE CH+454,4
    :POKE SC+454,160
    :POKE CH+458,4:POKE SC+458,
    160
635 IF POT>2 THEN POKE CH+476,4
    :POKE SC+476,160
    :POKE CH+480,4:POKE SC+480,
    160
640 IF POT>3 THEN POKE CH+453,4
    :POKE SC+453,160
    :POKE CH+459,4:POKE SC+459,
    160
650 FOR I=1 TO LEAF:READ P
    :POKE SC+P,65:POKE CH+P,5
    :NEXT:RESTORE
670 PRINT"[SPACE4,DOWN,RVS]
    HIT ANY KEY[RVOFF]"
675 IF POT>3 THEN PRINT"LARGEST
    POT"
680 GET K$:IF K$=""THEN 680
690 RETURN
700 IF LOC$="S"THEN
    LEAF=LEAF+INT(RND(1)*3+1)
720 IF LOC$="N"THEN
    LEAF=LEAF+INT(RND(1)*2)
730 IF LOC$="E"THEN
    LEAF=LEAF+INT(RND(1)*2+1)
740 RETURN
1000 IF DW<IW AND DF<IP AND
     DL<IL AND SIZE<IS THEN 1030
1020 GOTO 1040
1030 GOSUB 700:DW=DW+1:DF=DF+1
     :DL=DL+1:SIZE=SIZE+1
     :PRINT"PLANT GROWING"
1031 IF LEAF>BL THEN 2500
1032 GOSUB 600:GOTO 160
1040 IF DW>=IW THEN PRINT"PLANT
     TOO DRY":LW=LW+1
     :IF LW>3 THEN D$="THIRST"
     :GOTO 3000
1050 IF DF>=IP THEN PRINT"NEEDS
     FERTILIZER":LF=LF+1
     :IF LF>3 THEN D$="HUNGER"
     :GOTO 3000
1060 IF SIZE>=IS THEN PRINT"POT
     BOUND":LS=LS+1
     :IF LS>3 THEN
     D$="SUFFOCATION":GOTO 3000
1080 IF LOC$="N"AND DL>=IL THEN
     PRINT"PLANT GETTING LANKY"
     :D$="TOO LITTLE LIGHT"
     :GOTO 1100
1085 IF LOC$="E"AND DL>=IL THEN
     PRINT"PLANT IS UNHAPPY"
     :D$="TOO LITTLE LOVE"
     :GOTO 1100
1090 IF LOC$="S"AND DL>=IL THEN
     PRINT"PLANT GETTING SUNBURN"
     :D$="TOO MUCH HEAT"
     :GOTO 1100
1095 GOTO 1110
1100 J=RND(1)*10:IF J<2 THEN 3000
1105 GOTO 2000
1110 GOSUB 600:GOTO 160
1200 DATA 434,412,433,435,390,
     410,414,411,413,368,389,391,
     387,393,388,392,432,436,409,
     415
1210 DATA 367,369,346,431,437,
     408,416,386,394,366,370,345,
     347,365,371,324,344,348,430,
     438
1220 DATA 385,395,323,325,302,
     364,372,343,349,322,326,280,
     301,303
2000 GOSUB 600:LEAF=LEAF-1
     :IF LEAF<1 THEN D$="LOSS OF
     LEAVES":GOTO 3000
2020 GOTO 160
2500 PRINT"[CLEAR,DOWN,SPACE3]
     CONGRATULATIONS!","[DOWN3]
     YOUR PLANT BLOOMED IN","
     [SPACE5]"DAY" DAYS!!!"
2520 FOR I=1 TO 3000:NEXT
2530 PRINT"[CLEAR,DOWN,SPACE10,
     GREEN,SHFT A]"
2540 PRINT"[SPACE9,SHFT A,SHFT B,
     SHFT A]","[SPACE8,SHFT A,
     RED,SHFT Q,GREEN,SHFT A,
     SHFT N,SHFT A]","[SPACE7,
     SHFT A,SHFT M,SHFT A,SHFT B,
     SHFT A,RED,SHFT Q,GREEN,
     SHFT A]"
2550 PRINT"[SPACE6,SHFT A,RED,
     SHFT Q,GREEN,SHFT A,SHFT M,
     SHFT A,SHFT N,SHFT A,SHFT N]
     ","[SPACE5,SHFT A,SHFT M,
     SHFT A,SHFT M,SHFT A,SHFT B,
     SHFT A,RED,SHFT Q,GREEN,
     SHFT A,SHFT N,SHFT A]","
     [SPACE4,SHFT A,SHFT M,
     SHFT A,RED,SHFT Q,GREEN,
     SHFT A,SHFT M,SHFT A,SHFT N,
     SHFT A,SHFT N,SHFT A,SHFT N,
     SHFT A]"
2560 PRINT"[SPACE6,SHFT A,SHFT M,
     SHFT A,SHFT M,SHFT A,RED,
     SHFT Q,GREEN,SHFT A,SHFT N]
     ","[SPACE5,SHFT A,SHFT M,
     SHFT A,SHFT M,SHFT A,RED,
     SHFT Q,GREEN,SHFT A,SHFT N,
     SHFT A,SHFT N,SHFT A]","
     [SPACE4,SHFT A,RED,SHFT Q,
     GREEN,SHFT A,SHFT M,SHFT A,
     SHFT M,SHFT A,SHFT N,SHFT A,
     SHFT N,SHFT A,RED,SHFT Q,
     GREEN,SHFT A]"
2570 PRINT"[SPACE6,SHFT A,SHFT M,
     SHFT A,RED,SHFT Q,GREEN,
     SHFT A,SHFT N,SHFT A,SHFT N]
     ","[SPACE5,SHFT A,SHFT M,
     SHFT A,SHFT M,SHFT A,SHFT B,
     SHFT A,RED,SHFT Q,GREEN,
     SHFT A,SHFT N,SHFT A]","
     [SPACE4,SHFT A,RED,SHFT Q,
     GREEN,SHFT A,SHFT M,SHFT A,
     SHFT M,SHFT A,SHFT N,SHFT A,
     SHFT N,SHFT A,RED,SHFT Q,
     GREEN,SHFT A]"
2580 PRINT"[SPACE8,BLUE,CMDR *,
     RVS,SPACE3,RVOFF,SHFT POUND]
     ","[SPACE9,RVS,SPACE3,RVOFF,
     BLACK]"
2720 PRINT"[DOWN,SPACE3,RVS]
     TO PLAY AGAIN[RVOFF]"
     :GOSUB 670:GOTO 100
3000 PRINT"PLANT DIED FROM",D$
3020 PRINT"[DOWN,SPACE3,RVS]
     TO PLAY AGAIN[RVOFF]"
     :GOSUB 600:GOTO 100          C
```

VIC 20™
COMMODORE 64™

Still the Best!

Rated **THE BEST** educational program for the VIC 20™ by ***Creative Computing*** magazine.

Commodore 64 version: "This is the best typing tutor we have seen yet; it can get your children touch typing in short order and bring an old hand up to speed. Includes excellent training modules and an arcade type mode to liven things up and put some pressure on; ★★★★+" **INFO-64**

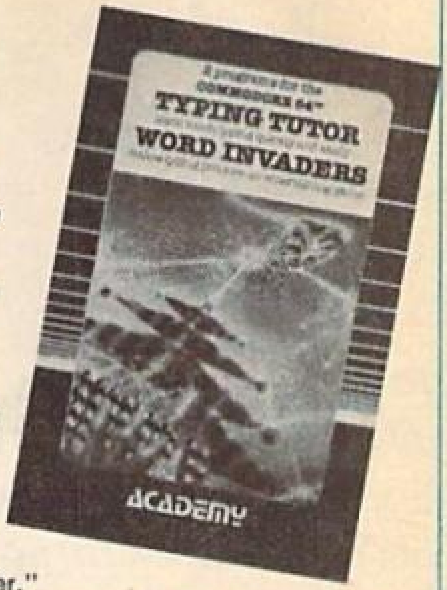
Our customers continue to tell us of their success. . . .

". . delighted with my son's progress . . . he is the only one in his second grade class who touch types at the computer."

(58 year old man writes) . . . "great, excellent. To me a source of great learning . . . I just can't express how much I have enjoyed it!"

In daily use by schools across the USA.

"Computer aided instruction at its best" ***Commander*** magazine



TYPING TUTOR + WORD INVADERS

The proven way to learn touch typing.

COMMODORE 64 Tape $21.95
COMMODORE 64 Disk $24.95
VIC 20 (unexpanded) Tape $21.95

NEW!

IFR
FLIGHT SIMULATOR
ACADEMY

IFR
(FLIGHT SIMULATOR)

DISK OR TAPE FOR THE COMMODORE 64
$29.95

CARTRIDGE FOR THE VIC 20
$39.95
JOYSTICK REQUIRED

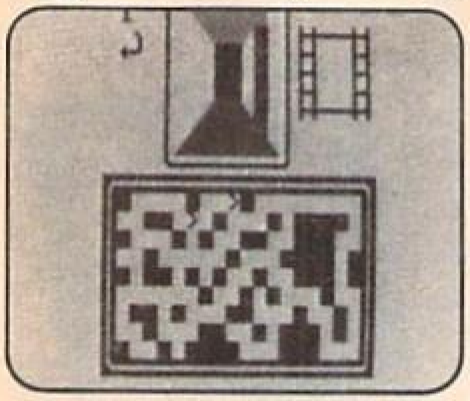
Put yourself in the pilot's seat! A very challenging realistic simulation of instrument flying in a light plane. Take off, navigate over difficult terrain, and land at one of the 4 airports. Artificial horizon, ILS, and other working instruments on screen. Full aircraft features. Realistic aircraft performance—stalls/spins, etc. Transport yourself to a real-time adventure in the sky. Flight tested by professional pilots and judged "terrific"! Rated "Excellent" by Midnite Software Gazette.

VISA Shipping and handling $1.00 per order. CA residents add 6% tax. MasterCard

ACADEMY
SOFTWARE

P.O. Box 6277, San Rafael, CA 94903 (415) 499-0850

Programmers: Write to our New Program Manager concerning any exceptional VIC 20™ or Commodore 64™ game or other program you have developed.



Shutter Bug

A 3-D maze game for the VIC 20 and Commodore 64

by C. D. Lane

You just graduated from the Phamous Photographer's School and landed a job with the Daily Times (though you moonlight for the Nightly Enquirer). Your first assignment is to enter the city's sewer system and photograph the giant ants reported to have nested there. Your assistant stays above ground with a radar unit, feeding you position information via radio which you track with your map. You are able to see your position and orientation, as well as the giant ant's orientation and position in the maze of pipes.

Along with the map, you see the view ahead of you in your view finder. When you reach an ant, you will see one of four possible views of him in your view finder—his head, right or left side, or his tail. Before you can take a picture you must wind your camera (press the space bar) else you will take a double exposure, possibly ruining a good shot. The Times pays more for better (i.e., head) shots. If you photograph the giant ant successfully, he disappears into the pipes and you must find him again (remember to wind your camera!).

There is an added twist to the VIC 20 version in that occasionally when you photograph the ant, he will steal your map! Now you must carefully follow his motions and turns in your view finder, following him through the maze of pipes. You must successfully photograph him again to get your map back (while he is blinded by the flash). It's not easy but it can be done. Carefully note what view of the ant you see when he disappears, then move forward and turn in the same direction if he turned or keep moving forward if his tail was the last thing you saw. Remember that the ant only turns when he can no longer move forward.

You have 24 frames, or five minutes, whichever comes first (usually the five minutes). Scoring is based on what shots you managed to get. You lose a slight amount for unshot or missed frames as an overhead fee (assistants with radar are expensive). However, the Nightly Enquirer will pay for your double exposures—but not enough to cover costs. The goal of the game is to get the highest dollar total you can.

The game requires a joystick (port two on the 64). Moving the joystick up moves you forward, down moves you backward. Moving the stick to the right turns you clockwise, left turns you counterclockwise. This may seem difficult to use at first but when your map is stolen, it is the only kind of movement that makes sense as you will not know which way you are facing.

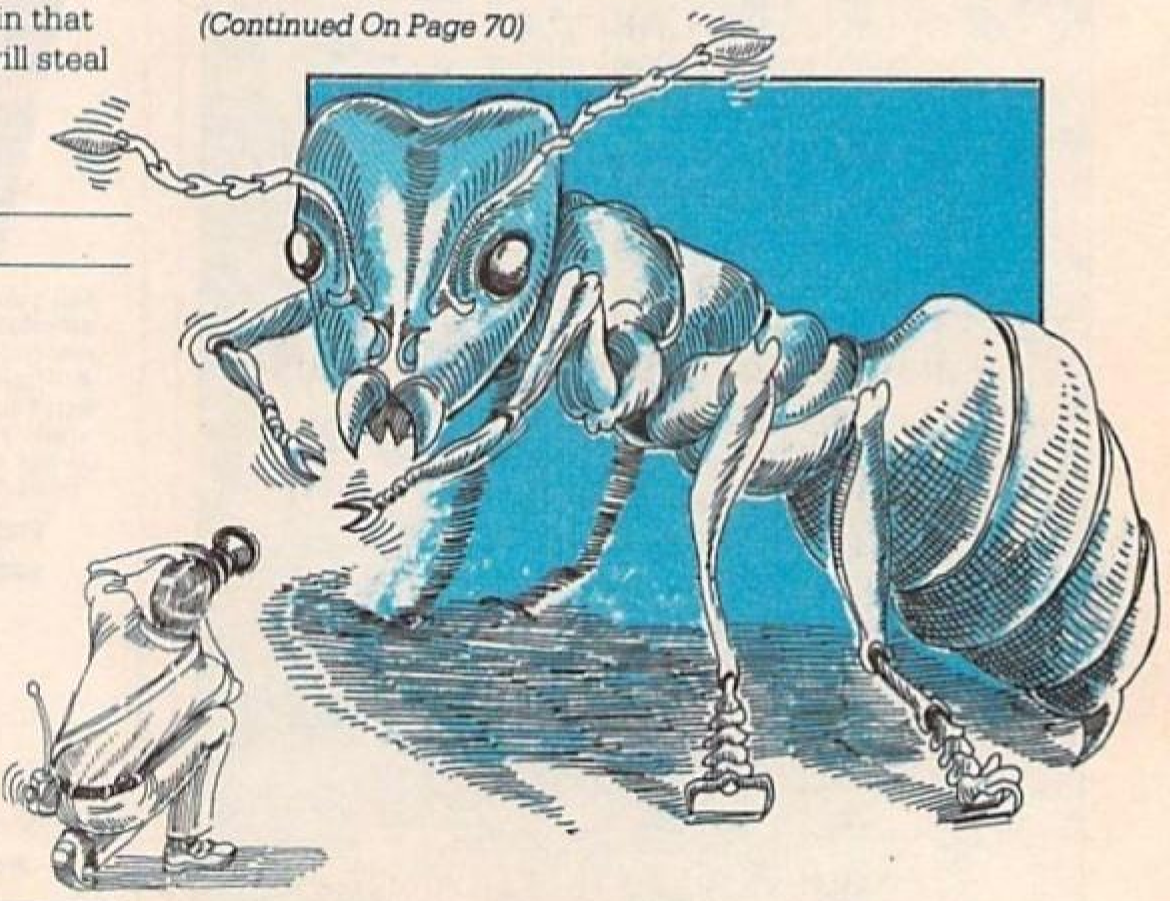
The maze is generated at random each time you play. Besides the maze and view finder, the screen shows the

(Continued On Page 70)

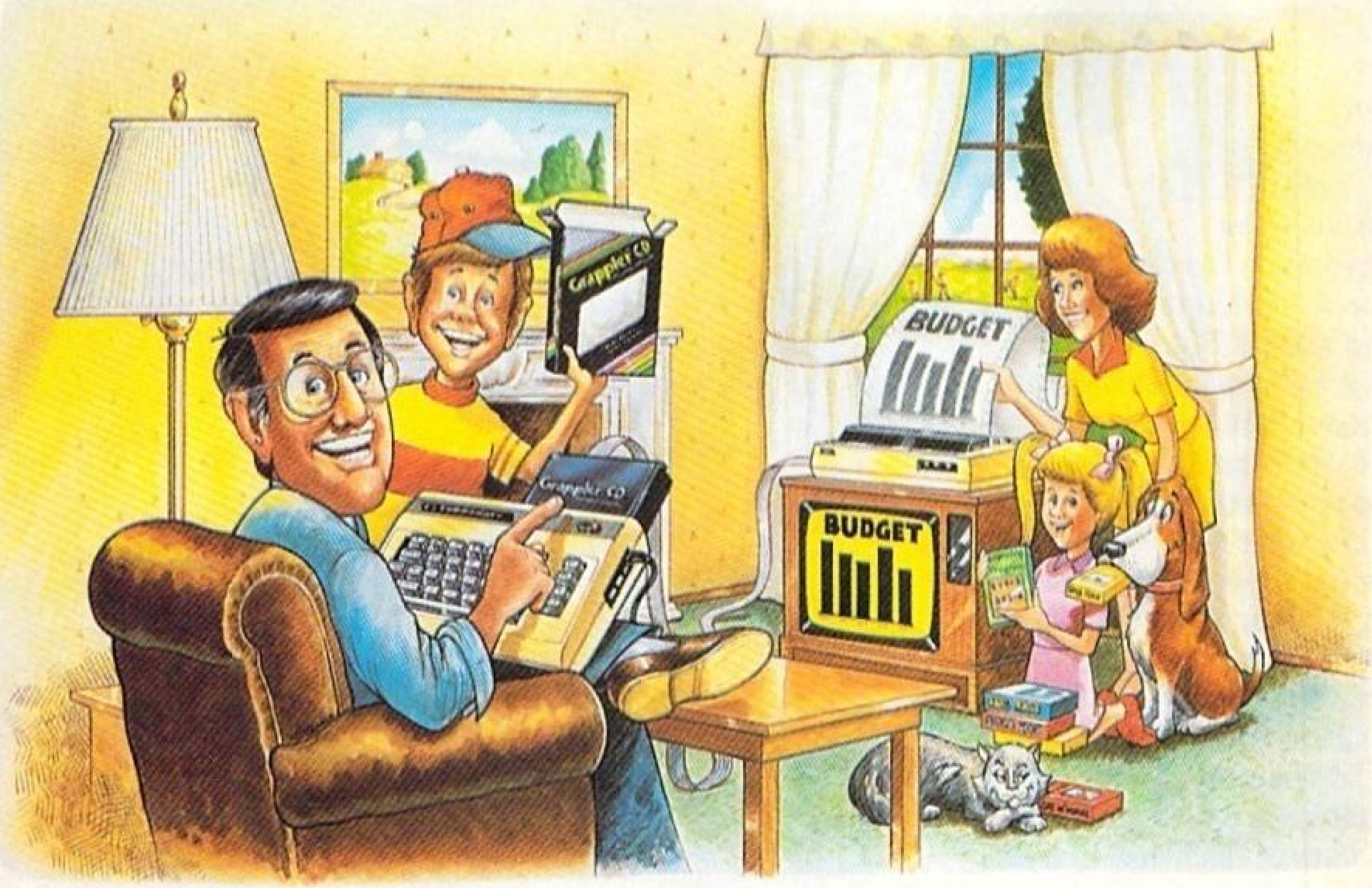
Implementation Notes

Map of the "Shutter Bug" program:

1	Array dimension initialization
2	Game timer
3- 8	3D view update
9-17	Joystick decoding
18-20	Giant ant motion and orientation
21-23	Main loop
24	Function definition
25	VIC 20 initialization
26	Commodore 64 initialization
27-35	Screen and variable initialization
36-40	Random maze routine
41-46	More initialization
47	Random maze location sub-routine
48-57	Camera "click" routine
58-64	Game end scoring
65	Data containing giant ant views



"Now Your Commodore 64™ Can Print Like a Pro!"



Grappler™ CD Printer Interface

The Revolutionary Printer Interface for the Commodore 64™

A New Era in Commodore Printing Power.

Grappler CD offers the first complete answer to your printer interfacing requirements, with many powerful capabilities unique in the Commodore marketplace. Complete signal translation allows many popular name brand printers to operate perfectly with the Commodore 64, or to imitate Commodore's own printer. Even Commodore's graphic character set can be reproduced on Epson, Okidata, Star, ProWriter and other popular printers.

Exclusive Grappler CD features provide a variety of graphic screen dumps, text screen dumps and formatting. No other Commodore interface can offer this.

If you own a Commodore 64…

If you're serious about quality, trouble free printing… You need the Grappler CD.

Contact your nearest Commodore dealer or call Orange Micro for a dealer near you.

Commodore 64 and Commodore 1525 are trademarks of Commodore Electronics Limited.
Epson is a registered trademark of Epson America, Inc.

A Uniquely Intelligent Interface:

- Prints Screen Graphics Without Software
- Graphics Screen Dump Routines Include Rotated, Inversed, Enhanced and Double Sized Graphics.
- Full Code Translation From Commodore's PET ASCII to Standard ASCII, the Language of Most Printers.
- Complete Emulation of the Commodore 1525 Printer for printing of Commodore's Special Characters.
- Dip Switch Printer Selection for Epson, Star, Okidata, ProWriter and other popular printers.
- Conversion Mode for Easy Reading of Special Commodore Codes.
- Text Screen Dump and Formatting Commands
- 22 Unique Text and Graphics Commands

Orange Micro Inc.

1400 N. LAKEVIEW AVE., ANAHEIM, CA 92807 U.S.A.
(714) 779-2772 TELEX: 183511CSMA

© Orange Micro, Inc., 1983

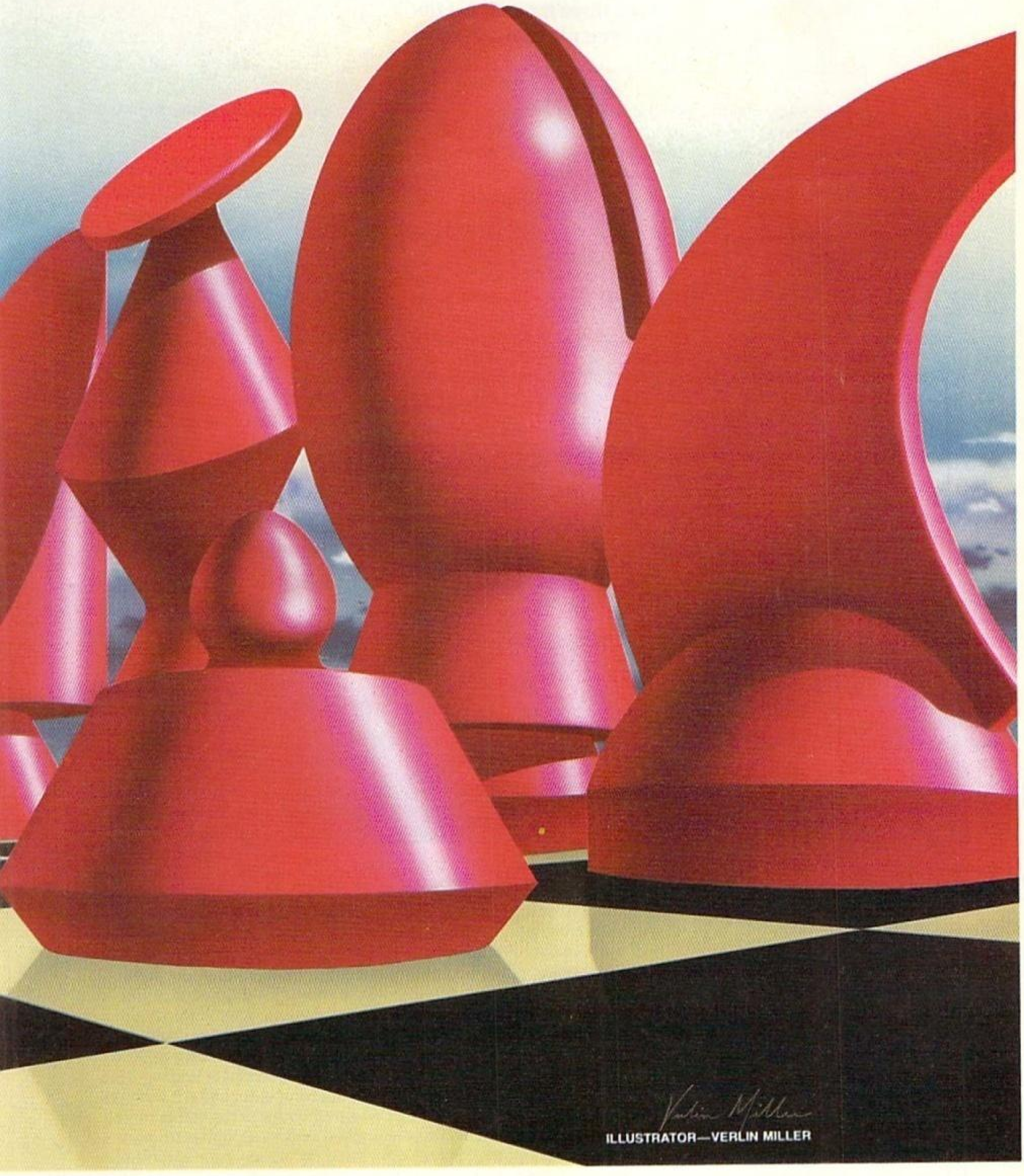
MORE THAN JUST

In January of 1983, Commodore announced the formation of a new Software Division and named Sig Hartmann as President. Since that time Commodore has emphasized the development of quality software and has, in fact, become the third largest producer of software (Wall Street Journal, February 23, 1984). Commodore Software's primary goal has been to produce a "complete menu" of high quality products at affordable prices. One hot section of the menu that has just started to be filled is that of strategy games.

By John Campbell
Commodore Software

A little tired of juicing up your joystick? Do you have calluses from taking on the Gorfian empire? Then prepare yourself for a new and exciting challenge. Commodore plans to release a group of new strategy games for the Commodore 64, starting this summer with its chess program. After examining hoards of chess packages for the Commodore 64 and other microcomputers, Commodore decided to acquire the rights to a modified version of the highly touted *Chess 7.0* from Odesta. Reviewers agree that the product is a state-of-the-art masterpiece. This program was chosen because it is much more than just another chess game.

ANOTHER CHESS GAME



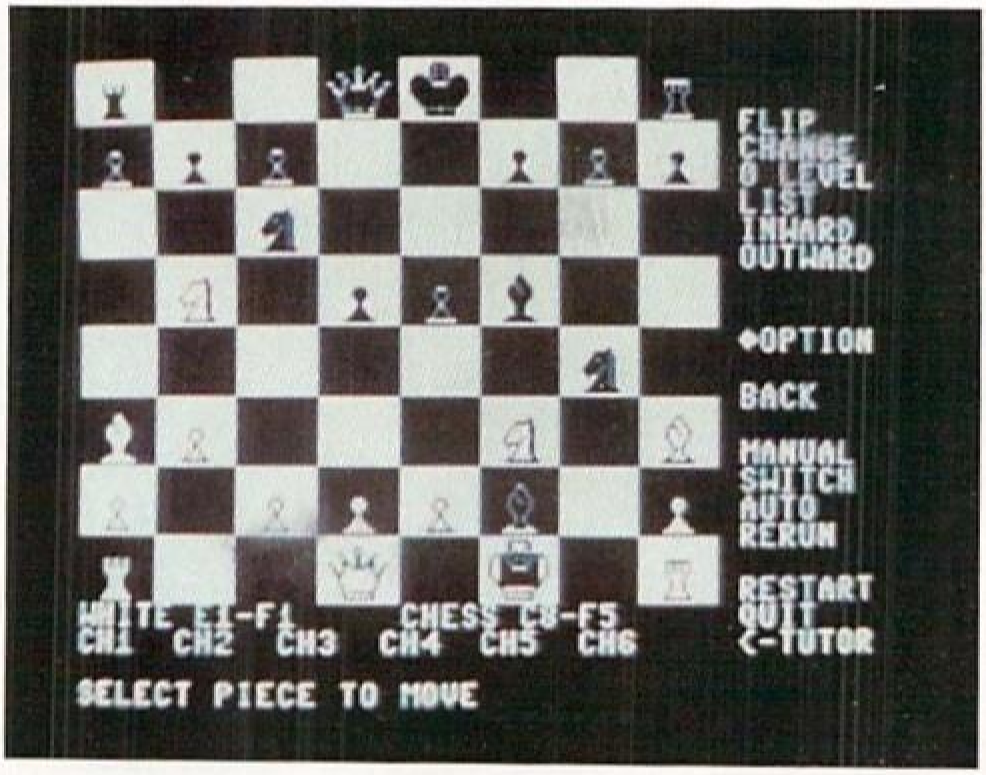
ILLUSTRATOR—VERLIN MILLER

When it comes right down to it, if you are serious about chess then you want to know if it is a tough competitor and if it will take three months for the computer to make a move.

Easy to Use. Unlike many other chess games, you can begin playing this game within minutes of opening the package. Simply load and run the program and wait for the playing board to come up on your screen. Since the program is rather extensive, the loading process takes a few minutes and this will give you time to start reading about the myriad of powerful features that are at your command. One of the reasons that the program is so easy to use is the screen layout (see Figure 1). With the playing board clearly represented, a versatile main menu, a message line for prompts and other information distinctly displayed, you do not need to know anything about computers (or standard chess notation) in order to enjoy a good game of chess.

The program begins with a demonstration of how to move the chess pieces and will continue until you press the fire button on the paddle (joysticks cannot be used with this program) or the RETURN key on the keyboard. (The fire button on the paddle and the RETURN key on the keyboard are basically interchangeable for the purposes of this game). The game difficulty defaults to zero, which is a good level for familiarizing yourself with the product. Making moves is very simple and the program will scroll through *only* the moves that are legally possible.

Tutorial. The fact that the game is easy to use is a critically important element. But the tutorial aspect of the program is what really sets the product apart from the rest. The program has six tutorial chapters in addition to the educational features in the main program. The six chapters are selectable in any order, and can even be picked during a game in progress if you need refreshing. The first three chapters are for the beginner, and they teach the rules of the game and some special situations (e.g., castling, *en passant*). The second three chapters are more strategy oriented and teach you how to use your pieces together in a cohesive plan rather than as just a bunch of individual pieces.



```
WHITE E1-F1      CHESS C8-F5
SELECT PIECE TO MOVE
```

The strong points of the tutorial are many. First of all, it is not just a text description. For example, when it is teaching the basic rules of the game it demonstrates the rules for each piece on the screen via a "movie". Another good aspect of the tutorial section is that in each chapter simple problems are provided to give you actual practice in given game situations (e.g., practice getting out of check). Finally, the tutorial is not aimed at just the novice. For example, chapter four demonstrates 12 different opening strategies and highlights the strengths and weaknesses of each.

Feature-packed. Most chess games are designed to handle the special moves like castling and *en passant*. Most chess games let you choose black or white and level of difficulty. Most games will also alert you to checks (or checkmate). But this chess game has numerous other valuable features (besides the tutorial section) that most chess games don't have, including:

- Advice: you can ask the computer what your best move would be.
- Auto: the computer will play both sides. It can be used at any time, even during a game. Helpful to learn some good defensive strategies.
- Inward: shows what pieces can attack a given chosen square (or piece).
- Manual: allows chess to be a two-player game, giving you the powers of the tutorial yet allowing you to play a peer rather than the computer.
- Rerun: replays all the moves made in a game so far.
- Switch: you can switch sides before or during a game, and the computer will be stuck in the situation you were in!

These are just a few of the powerful and unique features found in this chess product. But when it comes right down to it, if you are serious about chess then you are not that concerned about the tutorial capabilities or the special features. You want to know if it is a tough competitor and if it will take three months for the computer to make a move as it goes through some endless decision algorithm.

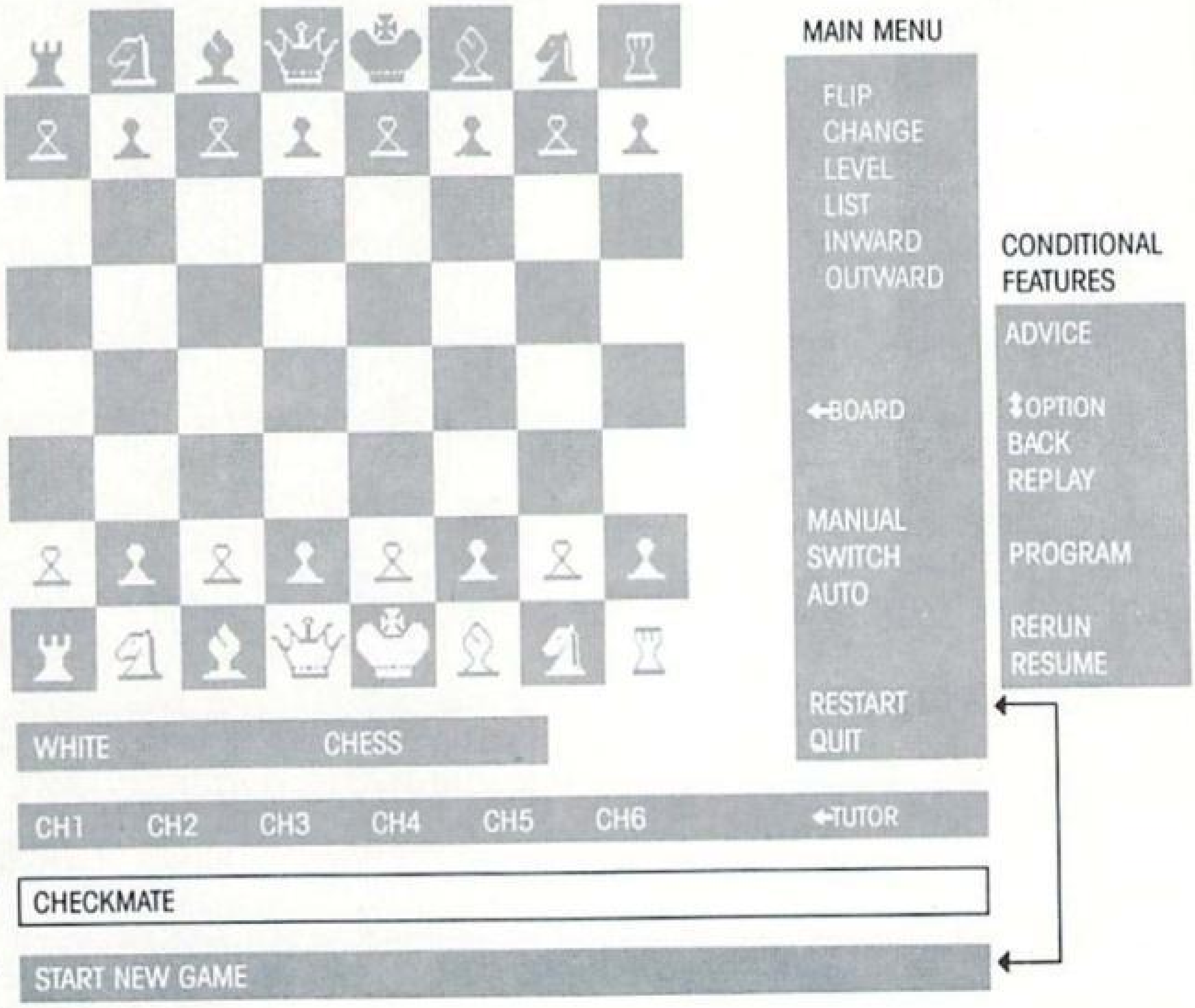
Difficulty and Speed. Predictably, the higher the level of difficulty chosen, the longer it takes for the computer to take its turn. But this chess product is surprisingly quick in its reaction time, regardless of the level chosen. At level zero, the computer responds almost instantaneously. At level five, the average time of response I experienced was about 30 seconds. Even at the most difficult level nine, the average response time was approximately five minutes. This compares favorably with some of the very strong chess games available today.

Unlike most microcomputer chess games, one notable facet of its combativeness is its strong end game. Just when you think you have crushed it, the computer turns the game around with a flurry of clever moves and leaves you devastated. Apparently the game lacks the ability to become frustrated when it is losing.

Concluding Notes. Although the game is extremely well-done, I'm sure some chess buff will think of some additional feature that would be nice to have. However, I couldn't think of anything that the program was lacking. The real point I would like to make is that even if you don't love chess, you will still enjoy this product.

Chess is the first of a group of easy to use strategy products that Commodore will introduce this summer. They are designed to challenge your intellectual concentration rather than your hand-eye coordination. If you're looking for a product that will entertain as well as educate, then Commodore's new chess product is for you. **C**

Figure 1. Screen Layout for Chess


MAIN MENU
FLIP
CHANGE
LEVEL
LIST
INWARD
OUTWARD
←BOARD
MANUAL
SWITCH
AUTO
RESTART
QUIT
CONDITIONAL FEATURES
ADVICE
↕OPTION
BACK
REPLAY
PROGRAM
RERUN
RESUME
WHITE
CHESS
CH1
CH2
CH3
CH4
CH5
CH6
←TUTOR
CHECKMATE
START NEW GAME

MOVE DISPLAY—Shows the coordinates for each move.

TUTOR MENU—Allows access to all tutorial features.

MESSAGE LINE—Shows checks, mate, prompts, etc.

DOCUMENTATION LINE—Shows description of each feature or each section of a tutorial chapter.

CAN YOU SURVIVE M.U.L.E.?

Start with some natural resources, food and money. Scramble for the best land grant, then build your net worth by developing and exploiting your resources. But remember, you'll have to cooperate with your fellow players, or nobody wins. This game of economics is both fun and educational.

Publishing Manager

M.U.L.E. is an easy to learn, fun to play, competitive strategy game for the Commodore 64. It lets up to four players participate in the same game but forces them to cooperate—if you fight too much, *no one* wins! In short, it is one of the best games *ever* for a computer, especially if you're weary of the standard "if-it-moves-kill-it" arcade games.

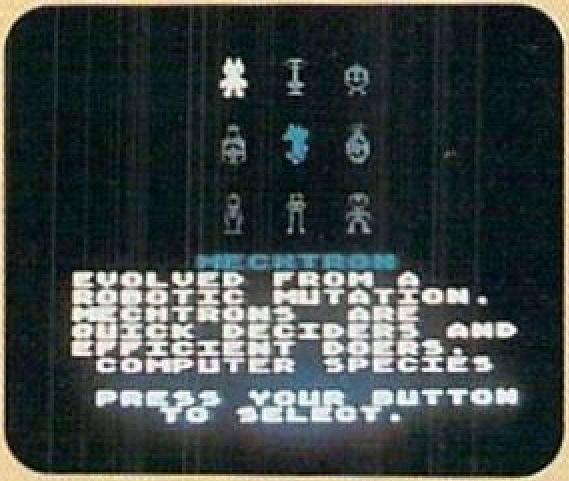
It is tempting to call *M.U.L.E.* a new kind of game—although it has roots in older economic games like *Hamurabi*—because it is the first game of the type that actually *works.* *M.U.L.E.* should do for computer economic strategy games what *Pacman* did for arcade action games, and I'm sure that the coming years will produce a whole school of games that "borrow" from the concepts pioneered here.

First of all, *M.U.L.E.* makes economics fun. The computer acts as your bookkeeper, which takes all the hard work out of the game. This is a big improvement over board games with similar themes, which make the players spend more time with pencil

ILLUSTRATOR—DAVID CHRISTIANA

M.U.L.E. should do for computer economic strategy games what Pacman did for arcade action games.

and paper than on the game play itself. You are free to concentrate on strategy, which is, after all, why you play the game.



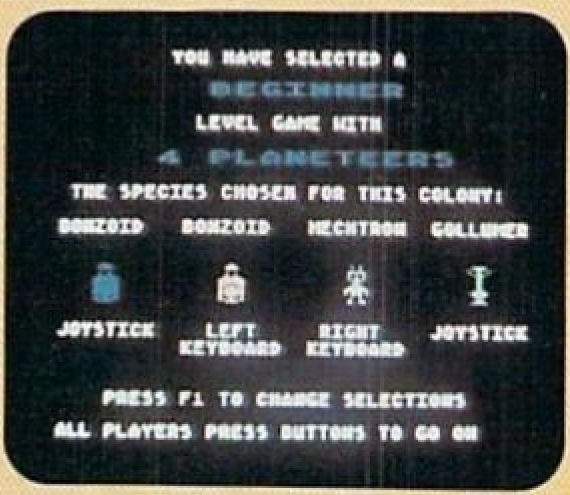
```
MECHTRON
EVOLVED FROM A
ROBOTIC MUTATION.
MECHTRONS ARE
QUICK DECIDERS AND
EFFICIENT DOERS.
 COMPUTER SPECIES

 PRESS YOUR BUTTON
    TO SELECT.
```

You can choose any of the eight characters with different abilities. The central figure is your choice, shown from the front, back and both sides.

M.U.L.E. is always a four-player game. If you have fewer players than that, the computer can take over the ones remaining. Each player starts with some natural resources, money, food and energy. The object of the game is to achieve the largest net worth value by the end of the game—six turns (at the beginners level) or twelve turns (standard or tournament levels). Net worth is built up by developing the industries of a new world, using the resources that are placed at random at the start of each new game.

The game includes a way to handicap players to balance the game play: there are eight different varieties of alien race you can choose to be, some with higher or lower amounts of money to start, some with different characteristics that affect their ability to produce certain commodities. So, for instance, you can let the stupid computer (playing the "mechtron") start with more money, give a novice human the ability to mine ore easier (the "bonzoid"), and start expert players with very little money (the "humanoid").

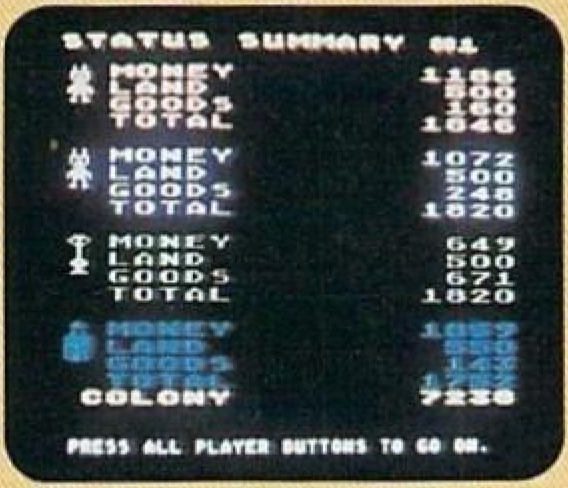
Like many strategy board games, turns are divided into several different phases. First comes the *land grant,* where players get to choose a free plot of land from the 44 plots on the world. The most interesting part of this is that it occurs for all players simultaneously—the first player to push the button on any plot gets it. In the tournament-level game, there may be a land auction where cash-rich players can bid for extra plots of land.



```
            YOU HAVE SELECTED A
                 BEGINNER
              LEVEL GAME WITH
              4 PLANETEERS
   THE SPECIES CHOSEN FOR THIS COLONY:
BONZOID   BONZOID   MECHTRON   GOLLUMER

JOYSTICK    LEFT      RIGHT    JOYSTICK
          KEYBOARD  KEYBOARD

     PRESS F1 TO CHANGE SELECTIONS
  ALL PLAYERS PRESS BUTTONS TO GO ON
```

Before the start of the action, the game shows everyone's color, character and keyboard or joystick choices.

The next phase is called *development*. One at a time, each player builds "factories" to create new commodities. To do this, the players walk their characters into the M.U.L.E. corral, obtaining a new M.U.L.E. at the going price. Next the M.U.L.E. is outfitted for production by entering one of the four stalls. Once outfitting is complete (a maddeningly slow process—M.U.L.E.s are stubborn!), the players walk to their plot and install the M.U.L.E., which becomes a factory for that item.



```
STATUS SUMMARY #1
  MONEY           1186
  LAND             500
  GOODS            160
  TOTAL           1846

  MONEY           1072
  LAND             500
  GOODS            248
  TOTAL           1820

  MONEY            649
  LAND             500
  GOODS            671
  TOTAL           1820

  MONEY
  LAND
  GOODS
  TOTAL
 COLONY           7228

PRESS ALL PLAYER BUTTONS TO GO ON.
```

Every round gives this status summary. Goods are valued at the average of the buying and selling price for that round. The colony total must reach $60,000 for anyone to win.

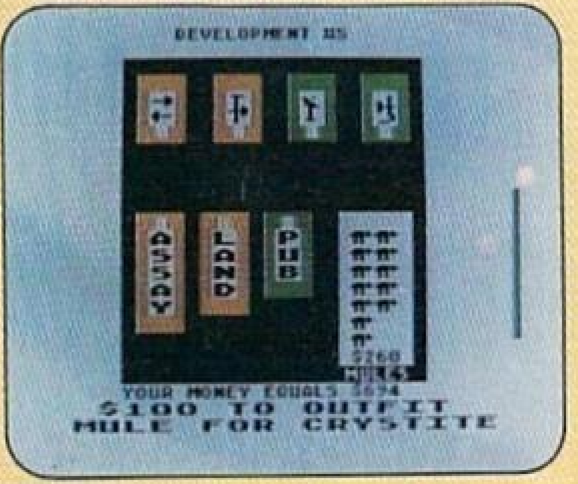
The economic laws of scale are important here, as are the basic resources available on a plot of land. Some plots are better at producing certain items than others (see Table 1.) The laws of scale add to the basic production values of each plot. If you have two plots of land next to one another producing the same item, each plot gets an additional unit of production. Also, any three plots you own producing the same item give each another extra unit. Location of spaces and what you develop on each (especially in the first few turns) are critical to your ability to prosper.

There are four different commodities that help you in different ways. Food gives you time to move during the development phase of the turn. Energy lets you achieve full production when farming or mining. Smithore produces more M.U.L.E.s, which are actually portable factories. Crystite is purely a cash crop, used outside the colony for jewelry or somesuch.

Once development is complete, the game moves to the *exploitation* phase. This is where the items are actually produced on the land. Items appear as rectangular dots on each plot. Certain events affect the levels of production, like earthquakes, which reduce mining by half, pirates who steal all the crystite, etc.

Next come the *auctions*. This is the heart of the game. Players can buy or sell any of the commodities, whose minimum prices are set by the laws of supply and demand. If there's a shortage of an item, the price will be high. The exception to this is crystite, which varies between $48 and $148 per unit randomly. It is to the advantage of the players to corner the market in an item (usually smithore) and let the price rise dramatically before selling. The only problem is that shortages slow down the growth of the entire colony, so you must cooperate sooner or later.

One nice twist to the auctions is the possibility of *collusion* between two players. Rather than let the price be determined by competitive bidding, a buyer and a seller can agree on a price and exclude anyone else. This lets the underdogs catch up and prevents the leader from gaining too great an advantage.

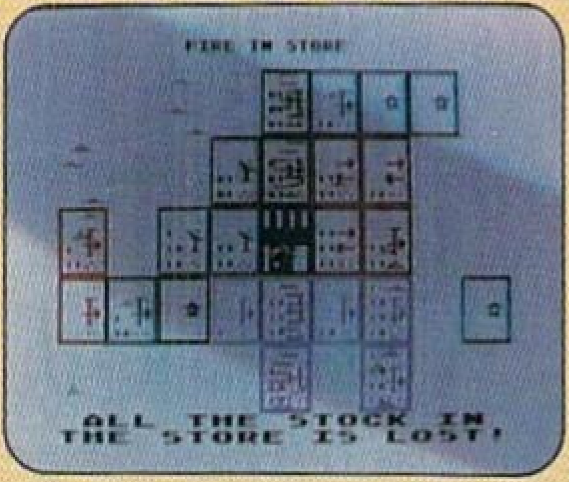


```
YOUR MONEY EQUALS $694
$100 TO OUTFIT
MULE FOR CRYSTITE
```

The bonzoid had picked up a M.U.L.E. from the corral at the lower right, then outfitted it with crystite from the upper right area. Now the M.U.L.E. is led to a plot of land to start producing.

Losing all your food due to a random disaster early in the game will just about wipe out your chances for success

After the auctions, the computer figures out each player's net worth and shows the four players ranked from top to bottom, with totals for the colony. By the end of the game you must reach $60,000 (tournament level) for anyone to win. Otherwise you all live—but poorly, in misery!


FIRE IN STORE

ALL THE STOCK IN THE STORE IS LOST!

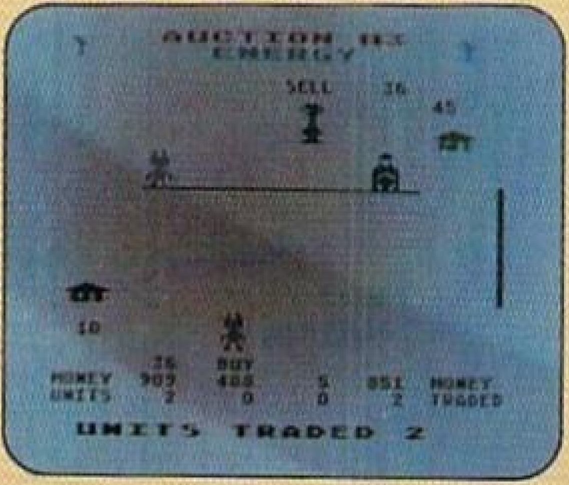
Each rectangle in the lower left of the plot is a unit of production. Random events like "fire in store" affect the supply of products available.

I haven't mentioned some of the events in a game of *M.U.L.E.*, like the random events affecting players before their development phases, but you get the idea that each game of *M.U.L.E.* is a diverse and entertaining experience. Don't let all the detail give you the idea that it is complicated, because it's easier than it sounds. A full tournament-level game takes only an hour to play. I usually get three games into my weekly session with my buddies Frank and Kevin (I promised to mention them as long as they keep letting me win).

There are many other nice features in *M.U.L.E.* The sound effects, especially the boppy little theme song, are professional and not irritating—you'll find that the sounds *help* you play the game. The graphics are adequate, if not spectacular, and the animation is very cute. The graphic system for auctions is perfect, and the mechanics for four players using two joysticks is well thought out.

The documentation and packaging for this game is a joy to behold. It is the most complete job of documenting a game I've ever seen. While the creators don't give the details on exactly how the mechanics of supply and demand are dealt with by the program (which would cause you to play using a calculator, which they were probably trying to avoid), details on the subtle aspects of the game are revealed. Nothing is left to chance. That's especially important in a strategy game like *M.U.L.E.*

If I had to I could pick out some things I didn't like about this game—losing all your food due to a ran-



The mechton, left, is buying energy at $17 from the flapper, right. The buyer moves up, the seller moves down, and when they meet they have a deal. The store sets a minimum price ($10) at which it buys and the maximum price ($45) at which it sells.

dom disaster early in the game will just about wipe out your chances for success—but it keeps me and my friends entertained for more hours than any computer game yet, including adventure games (and I've spent quite a while in those colossal caves, believe me). I expect to keep on enjoying this until someone manages to top it—and they've got some work to do!

By the way, this game is educational as well as fun. It gives you the clearest picture of the workings of economics available outside the business world. After mastering *M.U.L.E.*, you'll have a far better understanding of the workings of the real-world marketplace. *M.U.L.E.* would be helpful for social studies classrooms, as well.

M.U.L.E. is available on diskette for the Commodore 64 from Electronic Arts, for around $40. At last, a computer game worth buying a computer for!

Table 1: Production Abilities for Plots of Land

Type of plot	Food	Energy	Smithore
River	4	2	not allowed
Vacant	2	3	1
Mountain	1	1	2, 3, or 4

Hints for Better M.U.L.E. Play

1. Only play the tournament level game. Supply and demand aren't fully implemented in the other levels, and you'll just get confused trying to play what amounts to completely different games. The tournament game is the most fun, anyway.

2. The computer is a stupid player. It won't hoard precious items when it should. It also doesn't take advantage of the assaying process. It can find good crystite deposits but lets you exploit them. Watch the computer during the development phase!

3. Stop assaying after turn six. It wastes time. Just develop all the crystite you can. Even weak crystite plots will produce because of the economies of scale—and they'll help your good plots produce even more.

4. Don't sell energy unless you're desperate for cash! Each unit of energy you sell is worth hundreds of dollars in production to your opponents. Energy spoils slowly (one out of four spoils, rounded down).

5. Watch where you enter the town during development. If you enter from the sides you'll appear in the sides, but enter from the top or bottom edges and you'll appear in the middle. This can save you precious time.

6. Plan your production so that you are only mining by turn ten. You can usually buy energy and you won't need food by then. Switch river plots to energy and all others to crystite. **C**

ILLUSTRATOR—MARK PASCALE

for bridge players only

Whether you're an expert or a novice, you'll enjoy *Bridge 4.0* from Artworx for the Commodore 64. It's pure contract bridge that lets your computer be a partner, competitor and/or tutor.

By Joe Marlino
Commodore Education Department

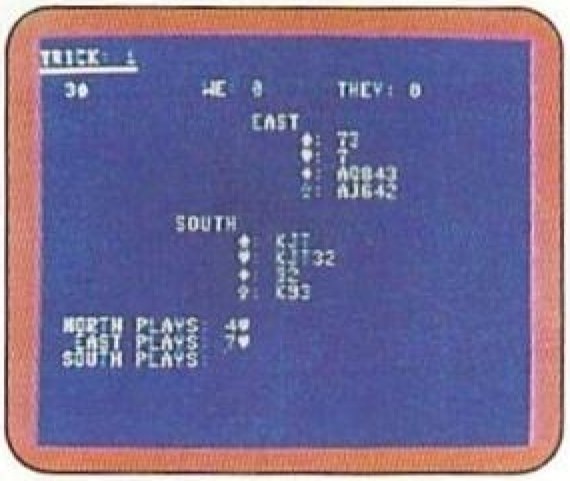
Having played bridge through my childhood with my family, I found *Bridge 4.0* by Artworx a challenge. It is an excellent way to learn to play bridge. Although it is geared for an experienced bridge player, it may be modified for the beginner. The novice should equip himself with a 3 × 5 card on which he has written the basics of the Goren Point Count System. The new bridge player may sit at his computer to play with three silent companions. The player has a computer partner and plays against two computer opponents. This is a much greater challenge than reading and studying the bridge hands published in the daily newspapers. There all four hands are exposed. It is a rare individual who can study his/her hand without looking at the other three.

The computer shuffles, deals and sorts the cards. The hand of the dealer—usually South—is revealed on the monitor. The player may study his hand at his leisure. After South bids, West bids, followed by North, and the bidding returns to the opening bidder, South. The bidding does not proceed until the player strikes the return key after each bid. The player may consider indefinitely West's bid. Also he may take his time in pondering what his partner's response signifies.

As the bidding proceeds a record of its sequence is kept on the monitor. There is no need to ask meekly for review of the bidding. After the contract is completed, the dummy's hand appears on the monitor, West leads and the game proceeds. As the tricks are gathered in, a record is kept on the top of the screen as to the number of tricks won by N/S and E/W. When the hand is completed the score is tallied and recorded on the monitor. The score returns to zero after a rubber is completed.

Another excellent teaching/learning device offered by *Bridge 4.0* is that the player is given the options of reviewing the hand and/or replaying it. Playing and studying the limitless number of hands offered by *Bridge 4.0* will help the bridge novice avoid some of the embarrassment often encountered by newcomers at the bridge table. "One spade," "One club," does not cause a nervous apology from your partner nor an annoyed glance from your opponents. The monitor merely records, "Error, bid again."

Bridge 4.0 is pure contract bridge. It is not a diluted version of the game in which you are presented with hands which have already been partially or completely bid. Nor do you have modifications of the rules which make it only a poor substitute for the game of bridge. Often in learning any game a player is limited by the playing ability of his opponents and/or partner. *Bridge 4.0* consistently provides realistic bidding and skillful playing of the hands.



```
TRICK: 1
 3♠            WE: 0       THEY: 0
                      EAST

                SOUTH

NORTH PLAYS: 4♥
 EAST PLAYS: 7♥
SOUTH PLAYS:
```

The viewer has the option of changing the screen, border and text colors by using the function keys on the Commodore 64. However, all card suits are in the text color. An obvious use of the Commodore 64's color graphics would be to display the heart and diamond suits in red and the spade and club suits in black. Another use of Commodore graphics would be to display the entire card. *Bridge 4.0* does not take advantage of Commodore 64's music and graphics capabilities. However, in a strategy game such as bridge these features are not needed to attract and maintain interest in the game.

The instructions for *Bridge 4.0* should include a pamphlet or tear-out card containing the Goren Point Count System and the basic rules of the game to aid the beginner. A list of bridge books could also be included in the instructions.

Bridge 4.0 is a challenge for bridge players at every level of the game from the beginner to the expert. The expert may return to a particular deal to test a different sequence of bidding or another game strategy. Also the computer game is an excellent way for partners to develop their bidding game and to discuss their methods of play. A lively discussion over the many hands available can result only in much better play at the bridge table.

Suicide Mission on the RUSSIAN FRONT

by Jim Gracely
Technical Editor

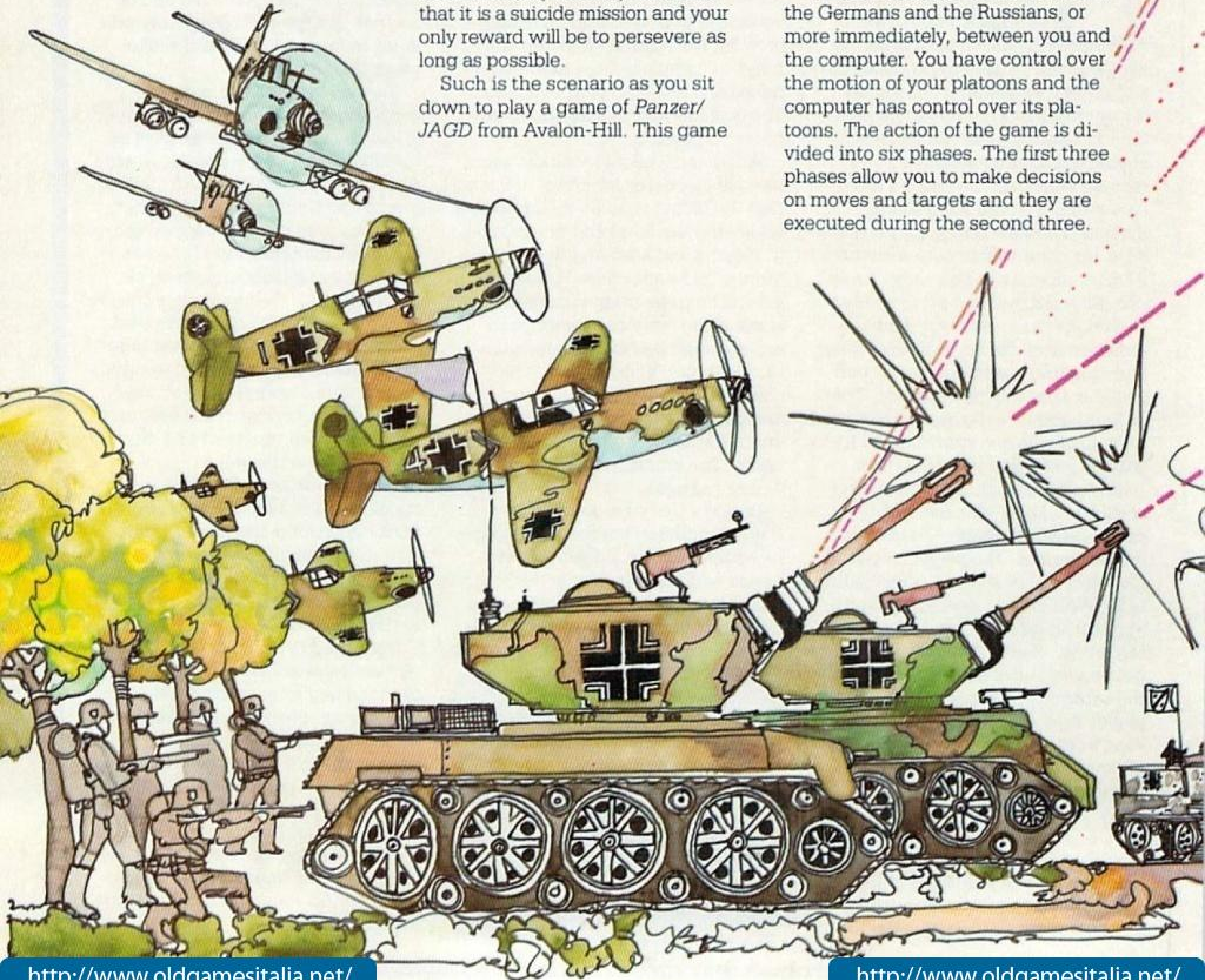
Relive the 1943 German offensive into Russia with Panzer-JAGD from Avalon-Hill Microcomputer Games. Chances are you won't make it back, but you've got to give it your best.

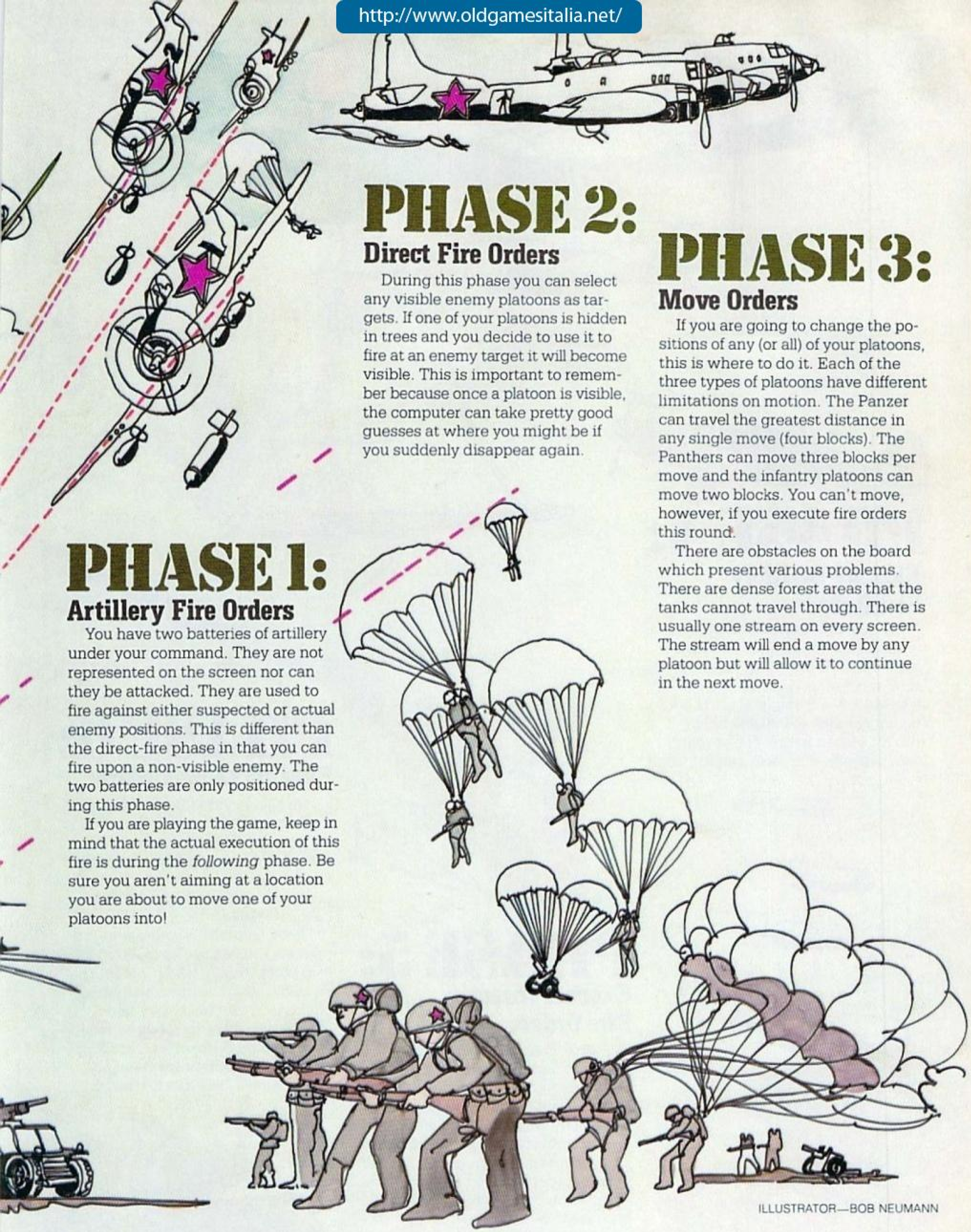
As games like *Archon* and *M.U.L.E.* propel us far into the future to tangle with the problems and monsters of that age, *Panzer-JAGD* throws us backwards to a time already history. Rather than postulate on the "what if's" of a time to come, we can play with the factors of the "what was" in a time past.

The year is 1943. You have under your direct command the left flank of the German front line offensive. You have at your command 12 platoons of the new Panther tanks (PzKw VD), four platoons of Panzers (PzKw III) and nine infantry platoons. You have been maintaining your position on the Russian front awaiting further orders when the command arrives ordering you to create a diversion from the main defensive. Your job is to advance your flank as far as possible into enemy territory. Odds are that it is a suicide mission and your only reward will be to persevere as long as possible.

Such is the scenario as you sit down to play a game of *Panzer/JAGD* from Avalon-Hill. This game is classed as a World War II tactical combat game. From what I can tell this means that everything is "by the book". The strengths, limitations and movements of each piece are based on actual data. The equipment that is used is also based on what was actually used by the Germans and Russians. There are no animate dead tank spells and no tank trolls hiding in the rivers.

This is a strategic game between the Germans and the Russians, or more immediately, between you and the computer. You have control over the motion of your platoons and the computer has control over its platoons. The action of the game is divided into six phases. The first three phases allow you to make decisions on moves and targets and they are executed during the second three.





ILLUSTRATOR—BOB NEUMANN

PHASE 1:

Artillery Fire Orders

You have two batteries of artillery under your command. They are not represented on the screen nor can they be attacked. They are used to fire against either suspected or actual enemy positions. This is different than the direct-fire phase in that you can fire upon a non-visible enemy. The two batteries are only positioned during this phase.

If you are playing the game, keep in mind that the actual execution of this fire is during the *following* phase. Be sure you aren't aiming at a location you are about to move one of your platoons into!

PHASE 2:

Direct Fire Orders

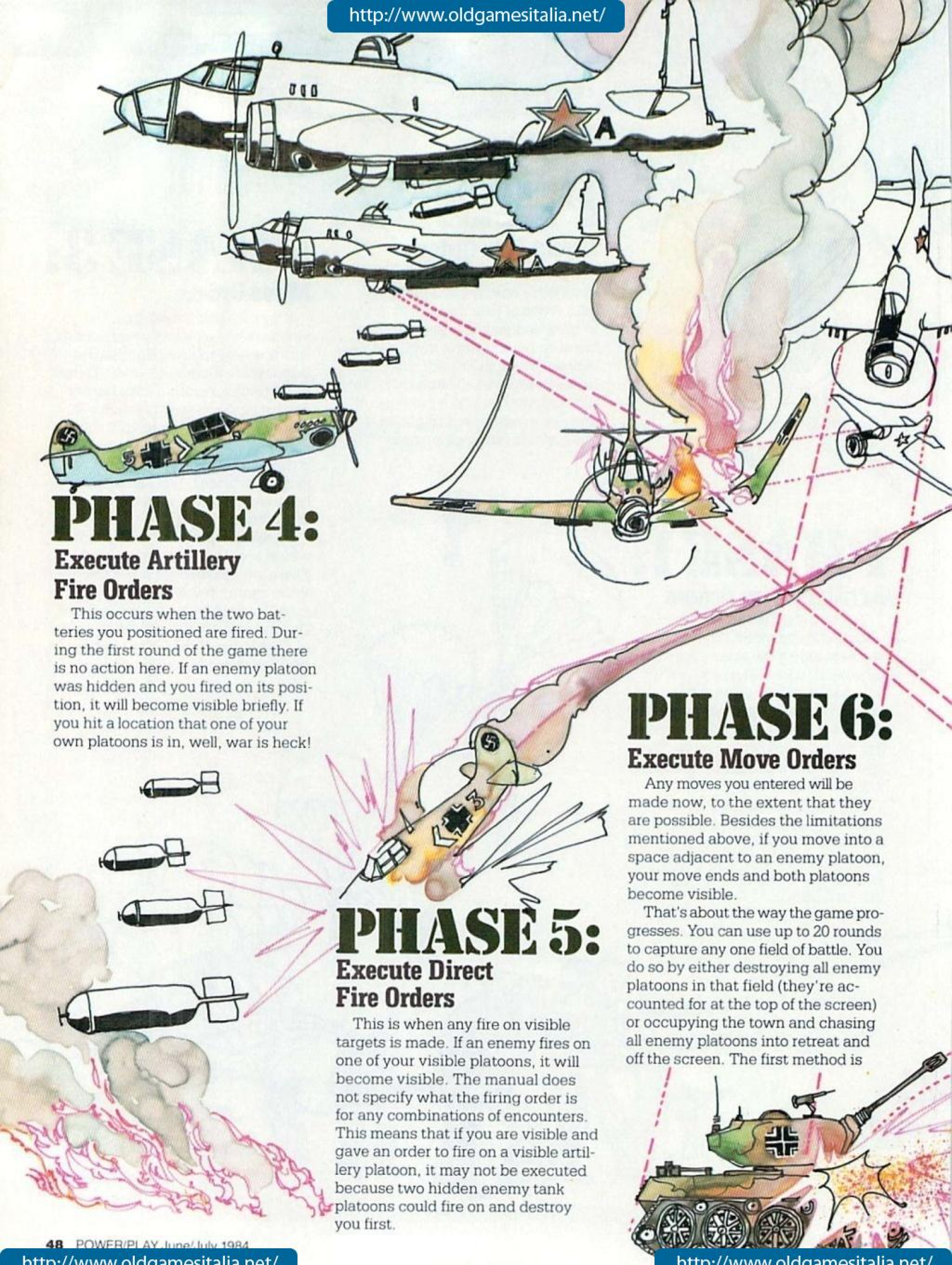
During this phase you can select any visible enemy platoons as targets. If one of your platoons is hidden in trees and you decide to use it to fire at an enemy target it will become visible. This is important to remember because once a platoon is visible, the computer can take pretty good guesses at where you might be if you suddenly disappear again.

PHASE 3:

Move Orders

If you are going to change the positions of any (or all) of your platoons, this is where to do it. Each of the three types of platoons have different limitations on motion. The Panzer can travel the greatest distance in any single move (four blocks). The Panthers can move three blocks per move and the infantry platoons can move two blocks. You can't move, however, if you execute fire orders this round.

There are obstacles on the board which present various problems. There are dense forest areas that the tanks cannot travel through. There is usually one stream on every screen. The stream will end a move by any platoon but will allow it to continue in the next move.



PHASE 4:

Execute Artillery Fire Orders

This occurs when the two batteries you positioned are fired. During the first round of the game there is no action here. If an enemy platoon was hidden and you fired on its position, it will become visible briefly. If you hit a location that one of your own platoons is in, well, war is heck!

PHASE 5:

Execute Direct Fire Orders

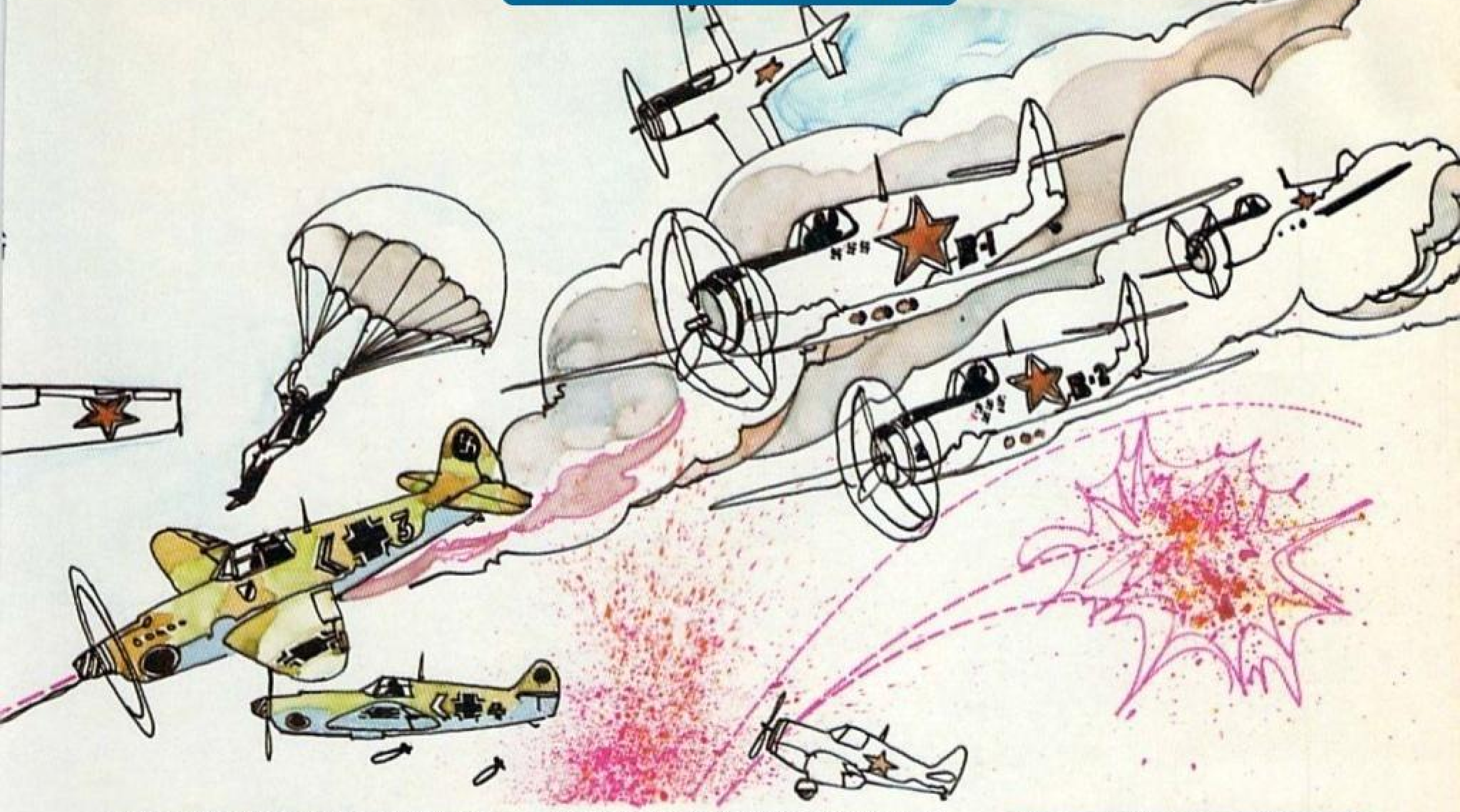
This is when any fire on visible targets is made. If an enemy fires on one of your visible platoons, it will become visible. The manual does not specify what the firing order is for any combinations of encounters. This means that if you are visible and gave an order to fire on a visible artillery platoon, it may not be executed because two hidden enemy tank platoons could fire on and destroy you first.

PHASE 6:

Execute Move Orders

Any moves you entered will be made now, to the extent that they are possible. Besides the limitations mentioned above, if you move into a space adjacent to an enemy platoon, your move ends and both platoons become visible.

That's about the way the game progresses. You can use up to 20 rounds to capture any one field of battle. You do so by either destroying all enemy platoons in that field (they're accounted for at the top of the screen) or occupying the town and chasing all enemy platoons into retreat and off the screen. The first method is

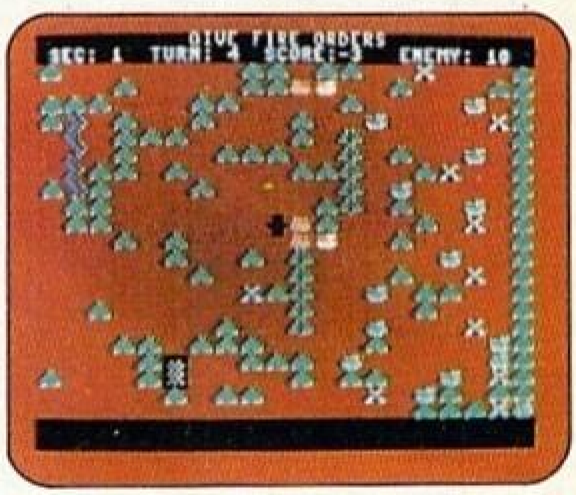


really preferable because all platoons you allow to retreat will be added to the platoons on the next battlefield.

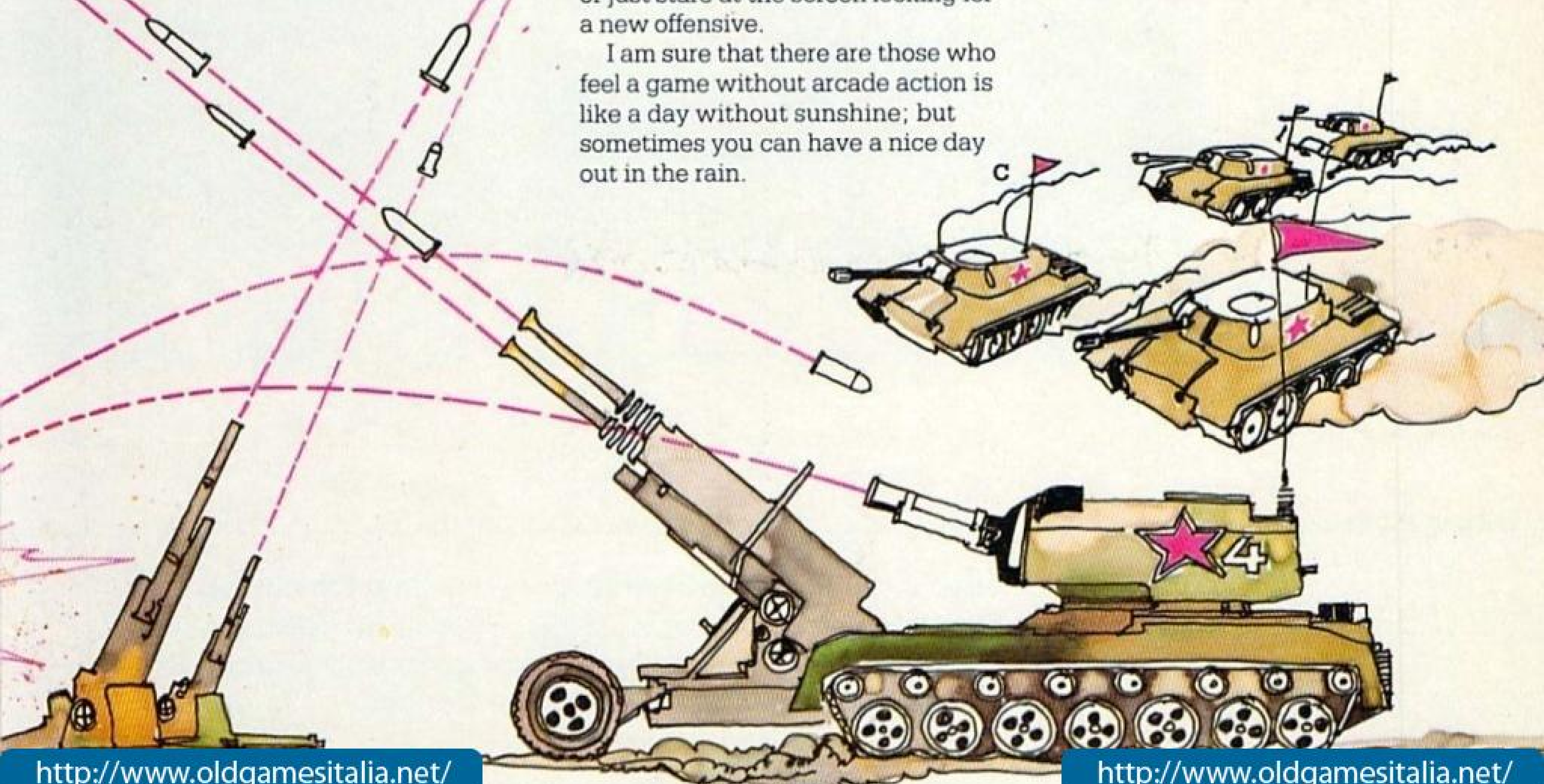
This is not a game to sit down and play for ten minutes while your eggs are boiling. A game played to the end (until you run out of platoons) can take many hours to play. By my third complete game I was getting pretty good at the strategies and it went on for about four hours. Of course, each time you play, the game lasts longer.

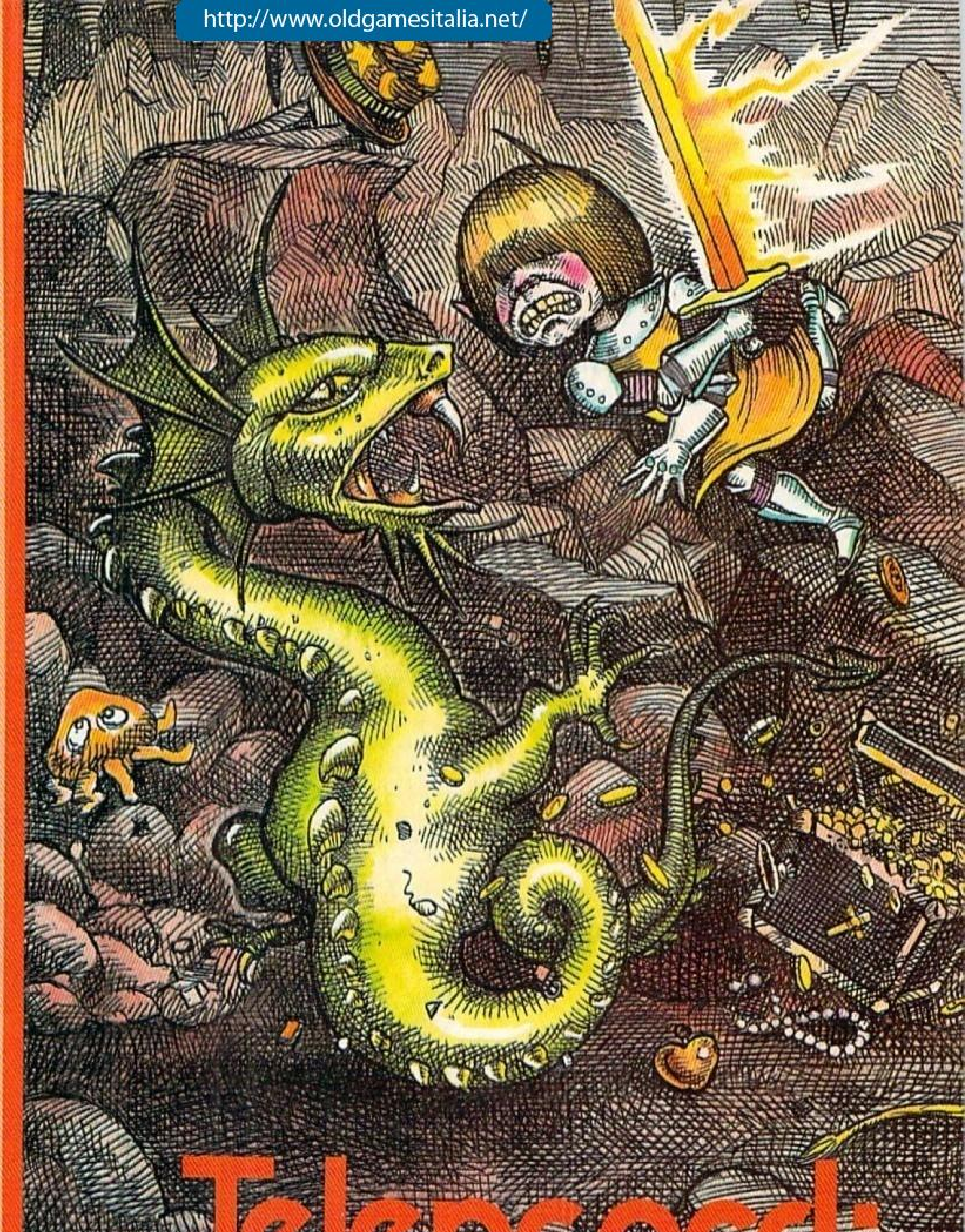
The real enjoyment of a game like this is realizing that there actually are strategies to use (both good ones and bad ones). Once you pick a strategy you can play it out as long as you like. This is a game of thinking rather than joystick agility. Between rounds there is no time limit and you can stop and have a sandwich, take a nap or just stare at the screen looking for a new offensive.

I am sure that there are those who feel a game without arcade action is like a day without sunshine; but sometimes you can have a nice day out in the rain.



```
GIVE FIRE ORDERS
SEC: 1  TURN: 4  SCORE:-3   ENEMY: 10
```

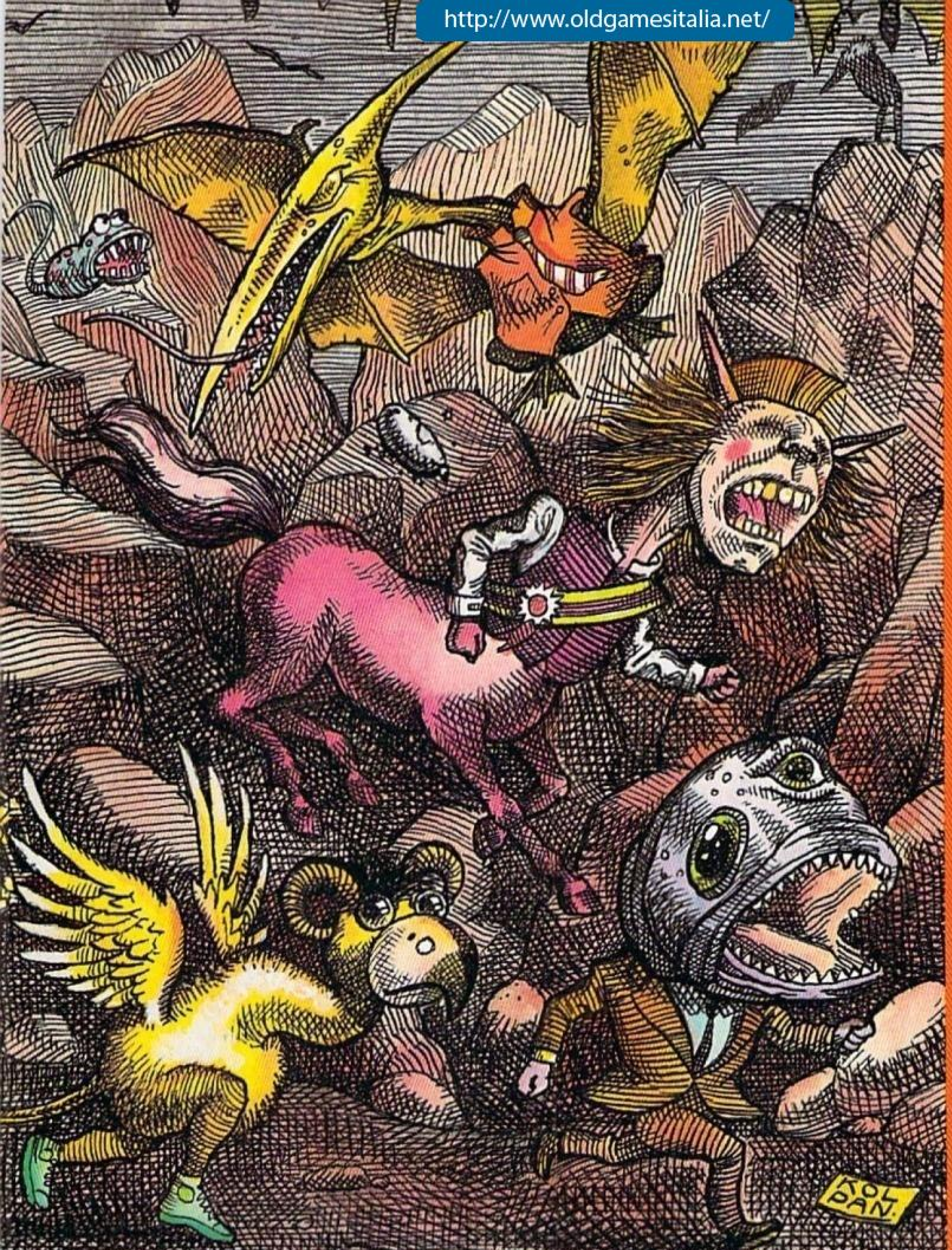




Telengard:

An Adventure in Terror

Descend into the labyrinth once again with our hero, Thrash, to fight dragons, trolls, ogres and vampires — and collect treasure chests full of gold and jewels. It takes strategy and luck to survive the halls of Telengard.



ILLUSTRATOR—GERALD KOLPAN

By Cyndie Merten with Bill "Thrash" Hindorff

One inevitably begins an adventure with Thrash somewhat in the dark. In this case, the darkness is real and terrifying. We are about to venture deep into the uncharted dungeons of *Telengard*.

The adventure begins by booting the *Telengard* disk from Avalon-Hill Microcomputer Games. We choose to become Thrash, a character who has been here before, and knows his way. His experience will be invaluable in pursuing the unknown treasures of *Telengard*, and in conquering the evil monsters who lurk there.

We begin at the bottom of a long staircase, with light (and civilization?) above. Thrash, never known to choose the easy way out, decides not to rise up, but rather to enter the dungeons. Suddenly, an ogre with level-four experience traipses in. "This ogre has been around awhile", says Thrash. We choose to fight (rather than evade or cast a spell). The outcome is victory; not a surprise to Thrash. Thrash gains some experience, but not much compared to his total. The refuse left behind by the ogre is worth one gold piece. This is a poor showing, but not bad for refuse, according to Thrash.

Traveling east one square, Thrash comes across a giant (level two) and gains 320 experience points by beating him up. While Thrash is standing there, a level-one elf appears. Thrash's body appeals to him, and the elf heals Thrash to his full strength. A level-one vampire comes along. Thrash casts a level-two spell and ensnares him in a web. Thrash proceeds to terminate him and receives 180 experience points.

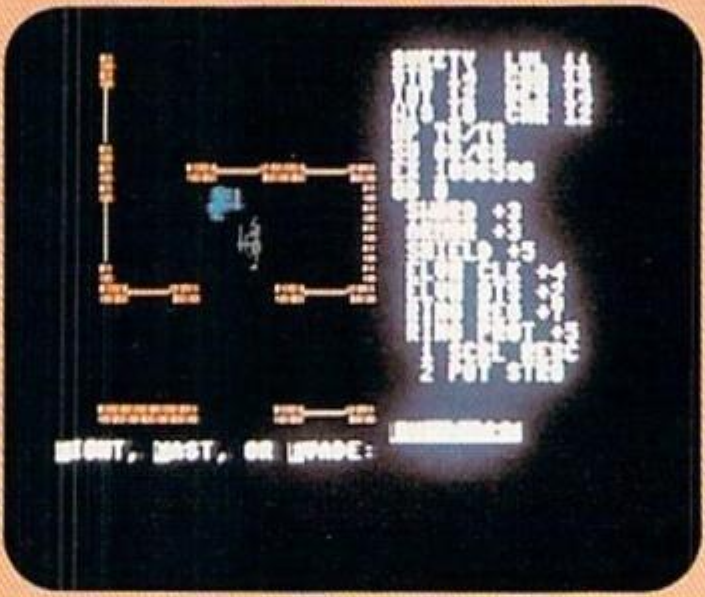
Proceeding north and west, Thrash comes across a level-three orc, which he fights and kills for 180 experience points. Going west again reveals a level-four ogre, which he kills. The ogre leaves behind a potion of strength, which Thrash grabs.

Thrash continues west and finds himself at a pit. Knowing the danger and risk involved in proceeding downward, Thrash will not risk our lives by doing so. "It is too difficult to get back out; we might not make it," he explains. We continue west to discover a large gray misty cube. We walk into it. It is a transporter to levels one through 50. Thrash hits one to try to stay at level one.

Alas, we have been teleported downward. Thrash knows because we encounter a level-24 demon (rarely seen near the surface). Thrash casts a level-six prismatic wall spell, banishing the demon and temporarily rendering us safe.

Ahead we see a jewel-encrusted throne emitting an eerie tune. A level-13 dragon appears. It repeatedly fries us with its fiery breath. We are damaged for 345, then 56, then 61 hit points. Finally, Thrash casts an ice storm spell to freeze the dragon. It dies. Saved again by Thrash's magic! Reading the runes increases Thrash's dexterity by one point. Sitting in the throne teleports us to another floor.

A level-38 giant appears. We must be deep inside the dungeon now! Thrash fights the giant twice to no avail. He casts a power word kill spell which is ineffective. But a plague spell finishes the giant off.



We head north and outfight a level-nine kobold. A mysterious elevator takes us up a floor. We come across a level-20 ogre, which Thrash fights and kills.

Thrash is tired from the fight. So many foes we have encountered this day. He drinks a healing potion recovered from an earlier fight. Now, he is ready to continue to guide me through the maze. He warns that we may be stuck deep in the dungeon for some time.

We come across a holy altar. A level-one vampire appears and is dispatched. A level-nine zombie is also dispatched, leaving behind a scroll of rescue. "This will allow us to leave when you are too tired to continue", Thrash informs me. Donating to the altar results in a hearty "Thank you", but no more.

Leaving the altar, we encounter a level-31 elf who restores the popular Thrash's body to full strength. Entering an elevator, we again rise a level.

"I think I grow too weary to continue", I say. "I still have not shown you a box", laments Thrash. A box has four buttons which, when pressed in the right order, open the box to reveal treasure. But, when the buttons are hit in the wrong order, the box zaps you with electricity.

"I beg of you to use the scroll of rescue", I say. But Thrash is unwilling. "It should only be used in an emergency. We will lose all our treasures."

Continuing to wander, we come across a box, at last. Thrash pushes the red button. "That was right". Again the red button. "That was right". Then green. ZAP!! electricity shocks Thrash. Undaunted, he proceeds to try more and more combinations. Finally it opens, revealing jewels worth 483,455 gold pieces. "That's why boxes are so good!" he says.

As we proceed we find a treasure chest with 108,453 gold pieces. Then, we come across a fountain with red water of which Thrash will not drink. But as we wait, the water clears, and drinking it heals ten hit points.

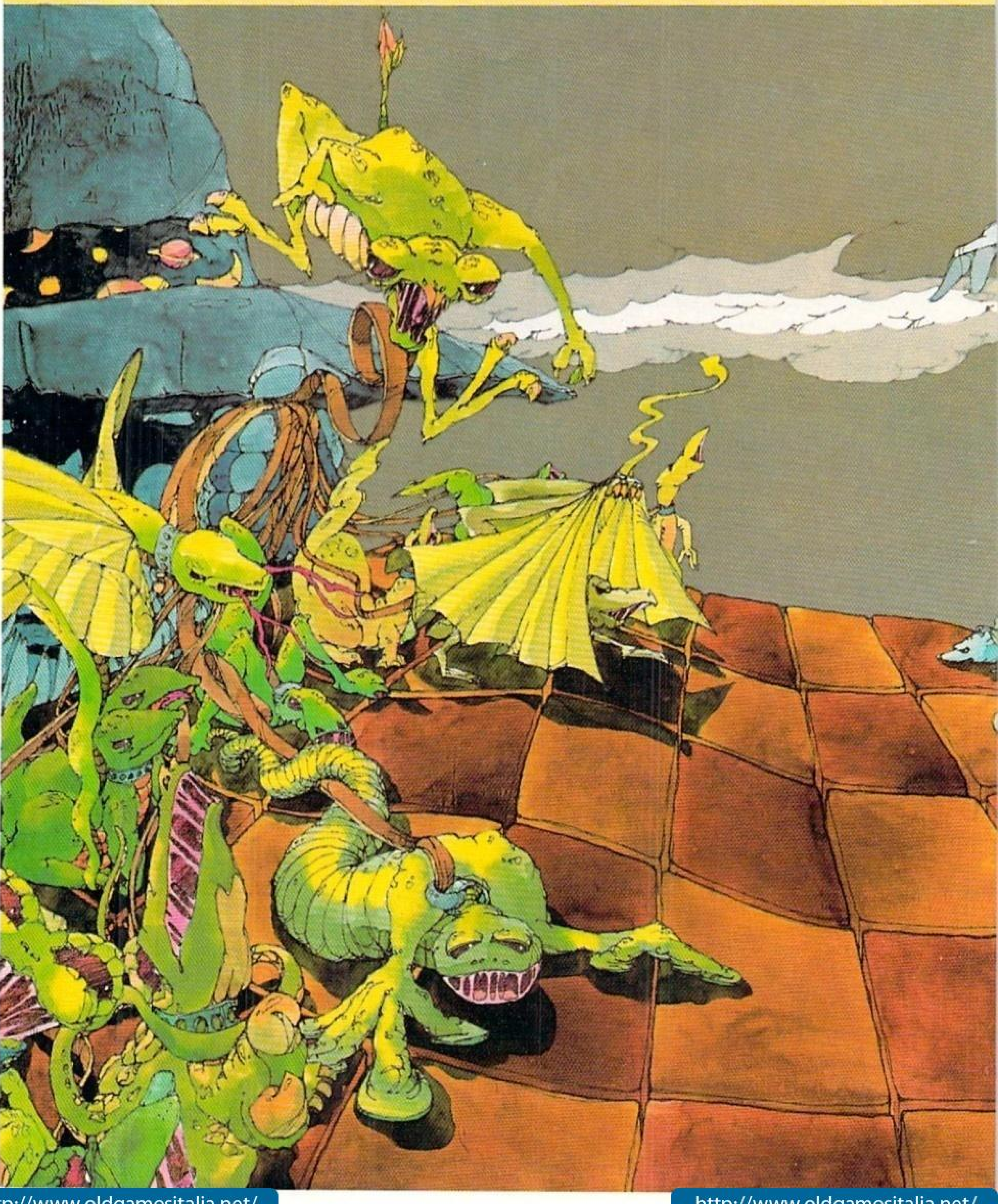
I have grown weary and Thrash agrees to try teleporting us out even though we may end up deeper in the maze. We teleport up a level. Teleporting again leaves our position unchanged. "We may be on level one", Thrash says hopefully. But, no, a level-14 dragon appears and begins breathing fire unmercifully. That does it. The tour is over. Thrash pulls out the scroll of rescue and

POOF!

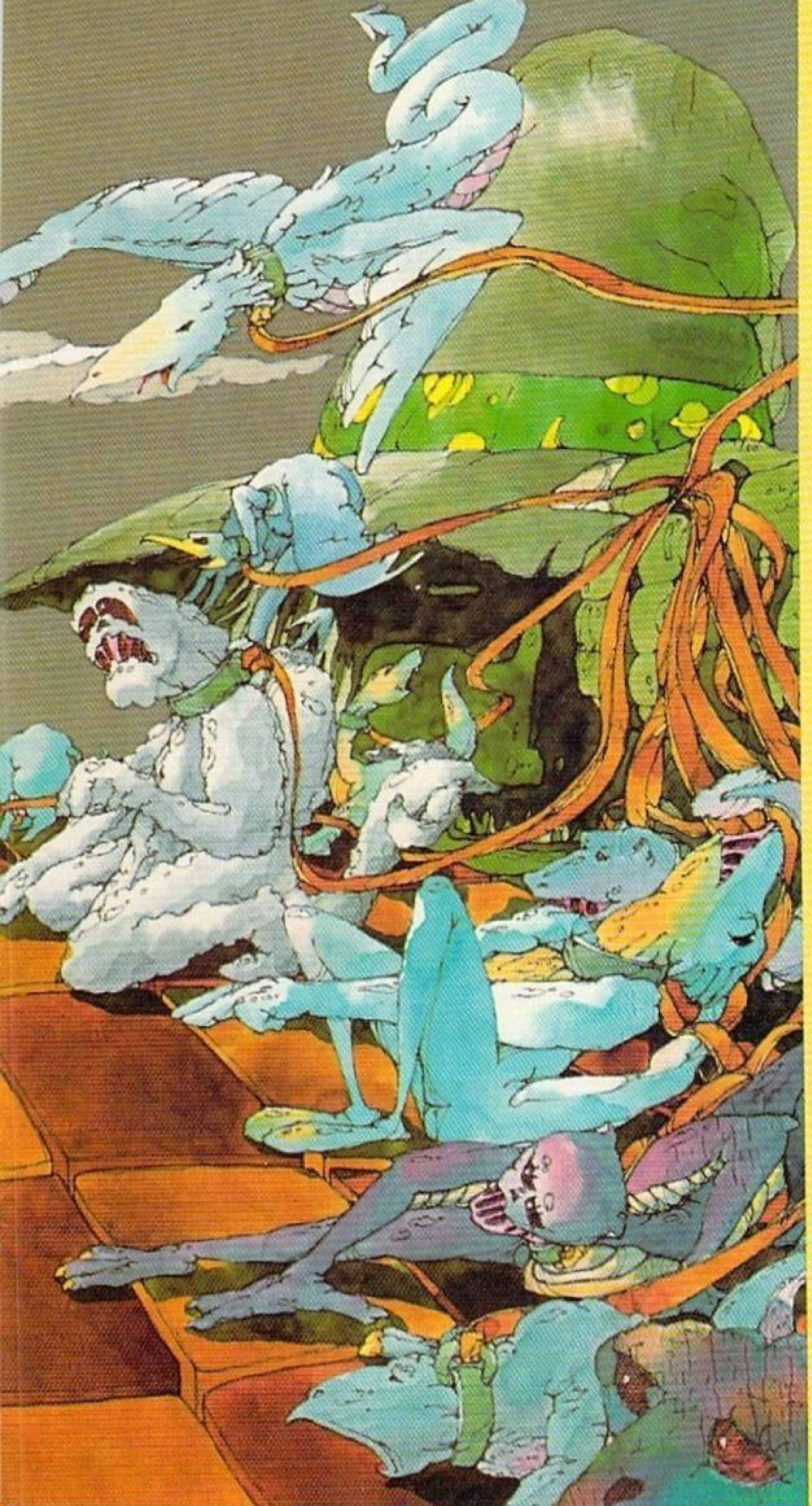
we are at the bottom of the steps. Our gold is gone, but the magic items remain. Thrash leads us to the Worthy Meade Inn for some rest.

Telengard is one of the deepest and involved dungeons ever encountered by Thrash. Hours of excitement and adventure await the fearless. The explorer may save his or her game at any point to continue another day. All in all, *Telengard* is a well-done graphic adventure which anyone would be proud to include in their program library. **C**

ARCHON:



By Eric and Tom Saberhagen
With Introduction By Betsy Byrne
Associate Editor



ILLUSTRATOR—GREG PURDON

CHESS, MAGIC & MONSTERS

The game that defies description. Is it chess? Is it an arcade game? Is it a fantasy role playing game? And can you ever win?? Only the people at Electronic Arts know for sure.

Have you ever speculated what it would be like to be part of a "living chess game", or even better, a participant not only alive but possessing magical powers? Then *Archon*, from Electronic Arts may be just your cup of tea. It is easier to describe *Archon* by telling you the type of game it *isn't,* since Free Fall Associates and Electronic Arts seem to have created a unique game—one that will not fit comfortably into any known genre of gaming.

Jon Freeman, a member of the Free Fall Associates design team that created *Archon*, has the best selling *Temple of Apshai* included in his impressive list of credits. In spite of its creator's background, however, *Archon* is *not* a fantasy role playing game or a simulation—although it certainly contains elements of each genre, and goes a long way toward satisfying the cravings of dyed-in-the-wool fantasy addicts.

Freeman is also known as an expert on board games, and is the author of *The Playboy Winner's Guide to Board Games.* You will find that *Archon* is more than a computer-style translation from a board—yet it contains elements of this time-honored genre as well. Possibly the best selling category of computer game is the arcade-style action game, and *Archon*'s three designers drew liberally from this genre—yet if hand-eye coordination is your only skill, *Archon* will surely defeat you. Strategy and tactics are important elements of *Archon*—to win you must become familiar with the strengths and weakness of each playing piece—but, of course, *Archon* is unlike any strategy game you have ever played. You get the idea. *Archon* doesn't defy *description*—just categorization.

Is *Archon* truly best described as a living chess game? There is no question that Freeman, Anne Westfall and Paul Reiche III drew on this idea when they began work on the game—it says so in the documentation that comes in the package. The idea for *Archon* is supposed to have come from Freeman's participation as King's Pawn in a game in which people became living chess pieces, fighting it out with other "chessmen" who invaded their board positions. Like the three dimensional holographic chess pieces in the movies *Futureworld* and *Westworld,* Freeman and his fellow pawns were commanded by players stationed well above the field of action, on a California hillside.

The first playing screen of *Archon* looks like a conventional chess board at first glance—but a closer look at the icons occupying the squares gives the first clue that all may not be quite as mundane as the landscape would lead one to believe. No tame little pawns or wily bishops on *this* board! Instead the stage is set for combat between the forces of Light and Darkness, Good and Evil, the powerful archetypical images drawn from our shared collective unconscious. The icons themselves are suspiciously close to the stuff that dreams are made of—or in the case of those on the side of the Dark—the cast of nightmares. Unicorns and Trolls, Mages and Evil Wizards, they all ride to battle at *your* command. Their weapons range from swords and spells to eye-beams and elementals. The players are moved first on the playing board in an attempt to capture the five power points on which they become invulnerable to all magic. Moving your piece to a square occupied by an opposing icon is interpreted as a challenge to battle: the scene zooms in to become a battlefield where you struggle to gain the advantage over your adversary, using the special characteristics of the hero or monster you command. You can choose to fight on the side of Light, or if you are feeling nasty, championing the Darkness may better suit your mood. Two players may compete, or the computer will agree to control the opposing force. *Archon* is probably *not* what most of us would imagine a living chess game to be like, but it seems to strike a chord in the depths of the least imaginative computer enthusiast, and for dreamers and idealists it will be an experience unparalleled in our mundane world of Pawns and Presidents.

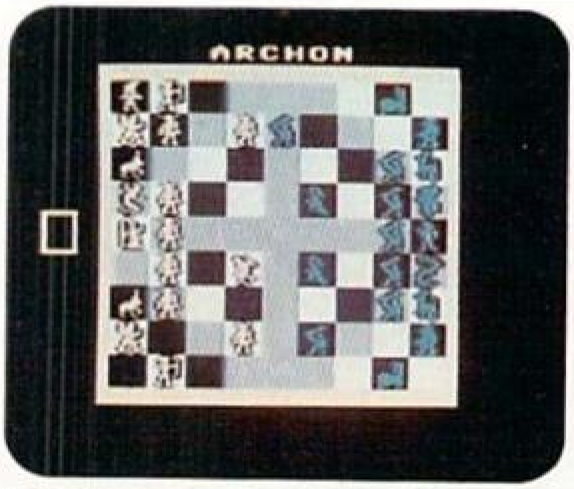
We asked Eric and Tom Saberhagen to venture into *Archon*'s universe and report to us on their experiences there. The two teenagers are tournament-class chess players and masters of both fantasy role playing games and strategy and tactics-type computer simulation games. Tom and Eric frequently review games for *K-Power* magazine, and have a well founded reputation as computer "whiz kids".

The boys' father is a well known science fiction author, and their mother designs and produces computer games based on the stories of science fiction writers. After several weeks with *Archon*, Eric and Tom (looking worn and disheveled) returned from their mission and delivered the following report.

The Game Examined

If you have ever wanted to be the general of an army of fantastic monsters, controlling them in battles to the death, you may be able to live out your fantasy with *Archon*. But take heed—*Archon* is not recommended for the faint of heart—danger and sorcery lurk here to trap the unwary commander. Your icons, computer simulations of creatures compiled from ancient bestiaries, fairytales and worse, will die messily and often as you strive to master this new game from Electronic Arts.

Archon may look like chess the first time you see the board, but try-



ing to capture an enemy piece will convince you otherwise. Someone has said, "There is no such thing as a free lunch." This old saw proves true in *Archon*, for you must fight for your lunch of trolls, dragon and other popular lunch meats. Monsters are not known for gourmet cooking, and if your monster, controlled by you with a joystick, is not fast and aggressive in the combat arena, he may wind up on some other monster's menu. Combat arena? Oh yes, when you attack an enemy beastie, the disputed square expands to fill the entire screen. It is in this arena that the outcome of the game is decided. If you can consistently win here, your victory is almost secured.

Filling the arena are many shrubs, planted by a deranged shrubber. These shrubs are on the disturbed side themselves, and they keep changing color. These form obstacles, which your monster must shoot and maneuver around. Successfully attacking a monster detracts from its lifespan, which is shown at one side of the screen. Reducing a monster's lifespan to zero kills it, and the victorious piece may claim the square.

There are two ways to win the game. First, all of the enemy's icons (better known as monsters) can be destroyed or imprisoned by magic. Second, all of the five "power points" can be occupied. "Power points" are small flashing centers where an icon is protected from all magic.

Magic can be cast only by the Dark Side's Sorceress or the Light Side's Wizard. There are seven spells that each magician can cast once. These spells range from summoning elementals to healing icons to imprisoning pieces.

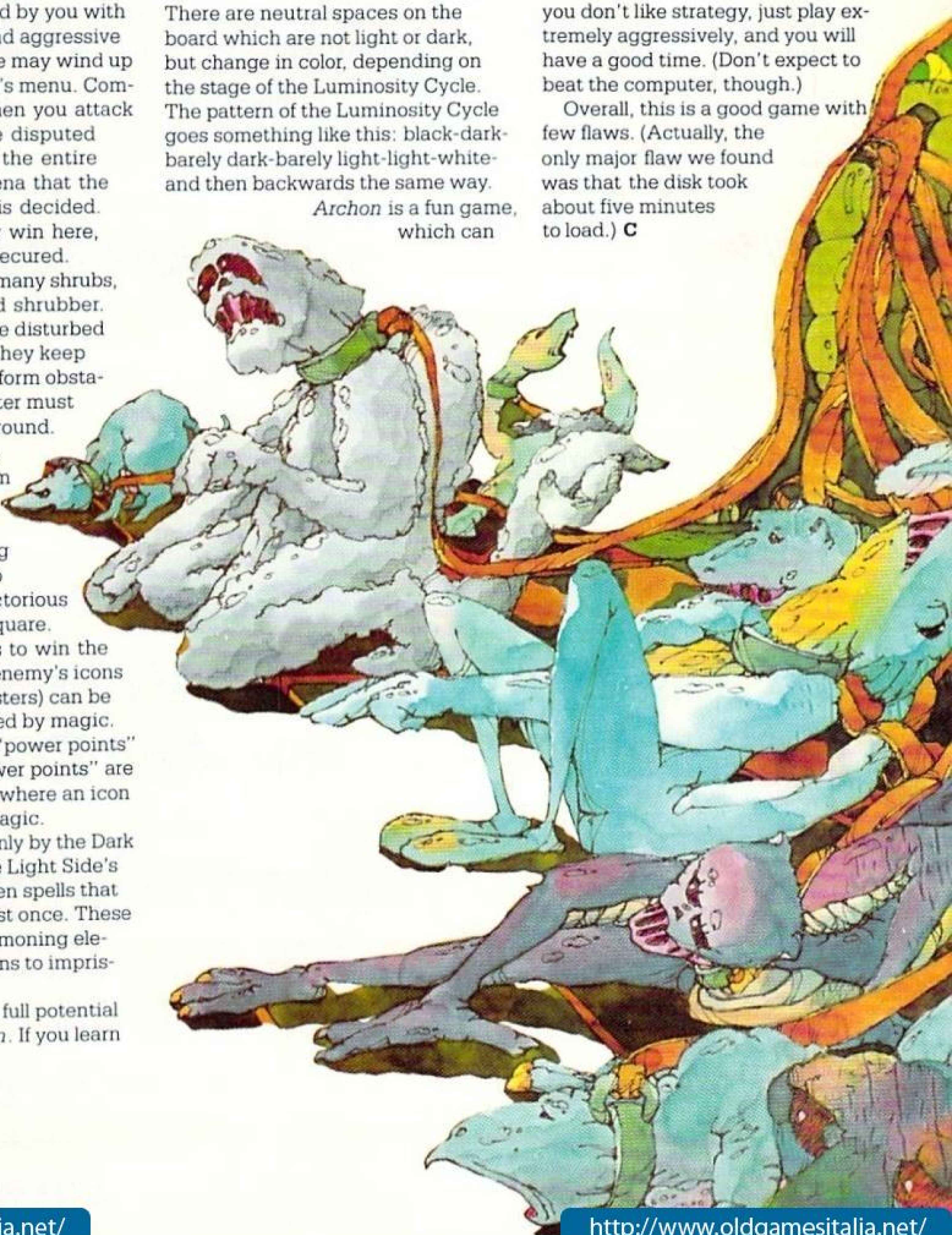
Using icons to their full potential is important in *Archon*. If you learn how to use a knight, you might have a chance at killing a troll with it. And if you learn how to use a unicorn you can beat almost anything with it. If you learn to use a Dark Side goblin, you can kill the most potent foes with it.

One more thing about playing. There are neutral spaces on the board which are not light or dark, but change in color, depending on the stage of the Luminosity Cycle. The pattern of the Luminosity Cycle goes something like this: black-dark-barely dark-barely light-light-white-and then backwards the same way.

Archon is a fun game, which can be played by either one or two players, but if you want to play with two players, you had better have two joysticks. If you don't want to play with two players, then be prepared for a tough game against the computer.

It is a good arcade-style game with strategy in it, too. Still, even if you don't like strategy, just play extremely aggressively, and you will have a good time. (Don't expect to beat the computer, though.)

Overall, this is a good game with few flaws. (Actually, the only major flaw we found was that the disk took about five minutes to load.) **C**



WE LOVE COMMODORE
and
We Love Our Customers

That's why we only sell and support Commodore 64 and Vic 20 computers!! We have • the best prices • over 1000 programs • 500 accessories • absolutely the best service • one day express mail delivery • immediate replacement warranty • 15 day free trial • programming knowledge • technical knowledge • we are the only one in the U.S.A. with complete support for Commodore 64 and Vic 20 computers!!

PROTECTO ENTERPRIZES Box 550, Barrington, IL 60010

SEND IN THIS COUPON TODAY FOR A FREE CATALOG AND OUR SPECIAL
"BUY MORE—SAVE MORE COUPON"
(Save up to $500 on software and accessories)

Name ____________________

Address ____________________

City ____________ State ________ Zip Code ________

No One! But No One! Can Compare

TO

PROTECTO ENTERPRIZES

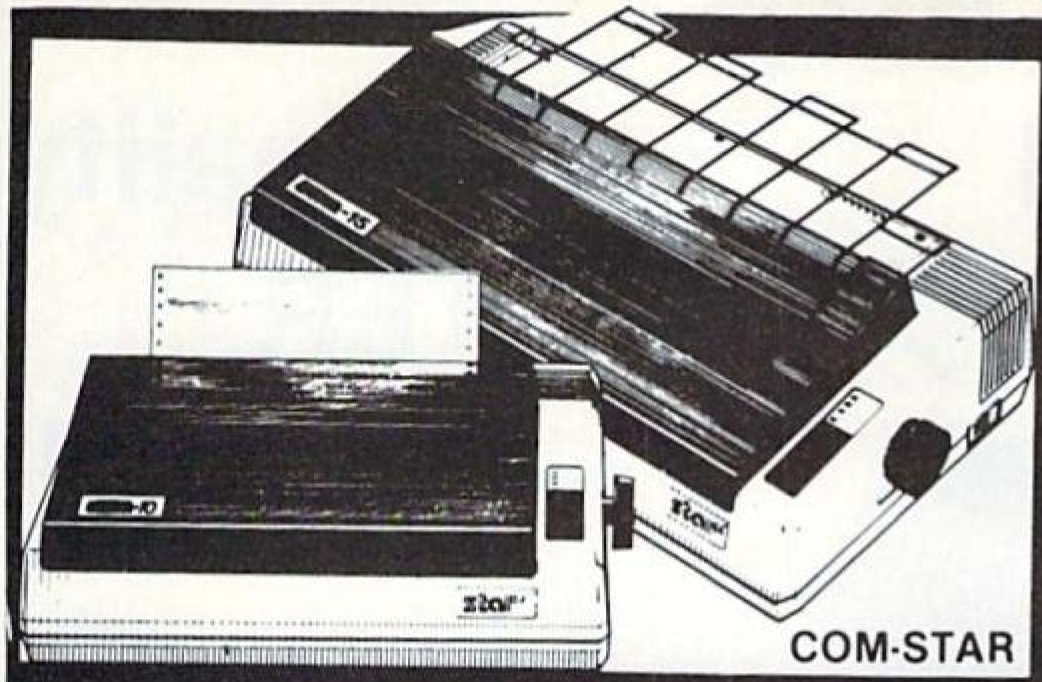
TO ORDER WRITE OR CALL: PROTECTO ENTERPRIZES, BOX 550, BARRINGTON, IL 60010

Call 312/ 382-5244

8 to 5 Weekdays 9-12 Saturdays

(See Next 11 Pages)

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor Friction Printer

only $199**

•15 Day Free Trial -180 Day Immediate Replacement Warranty

- Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.
- Fast 80-120-160 Characters Per Second
- 40, 46, 66, 80, 96, 132 Characters Per Line Spacing
- Word Processing
- Print Labels, Letters, Graphs and Tables
- List Your Programs
- Print Out Data from Modem Services
- "The Most Important Accessory for Your Computer"

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$199.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$429.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$529.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64— $69.00

For All Apple Computers—$79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

4 Color

80 COLUMN Letter Quality PRINTER / PLOTTER

Super Special

$99

1/2 PRICE SALE

1520
commodore

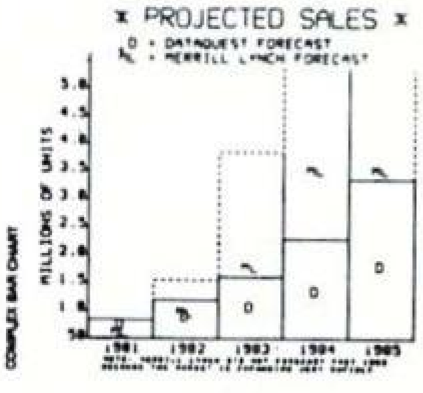
$99

1/2 PRICE SALE

- LOWEST PRICE IN U.S.A.

Commodore-64 & VIC-20

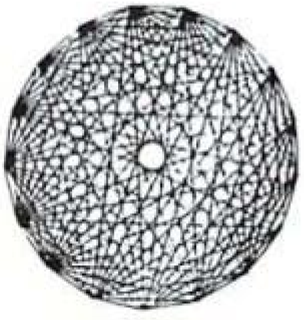
• List your programs • High resolution graphics for bar charts and geometric figures (like spirograph) • Plugs directly into VIC 20 and Commodore 64 — Interface included • Lowest cost letter quality printer in the country.



ACTUAL PRINT SAMPLES

UPPER CASE – ABCDEFGHIJKLMNOPQRSTUVWXYZ

LOWER CASE – abcdefghijklmnopqrstuvwxyz



At last you can list your programs (even control characters) and make beautiful high resolution graphics at an affordable price. This **80 column letter quality printer/plotter** is great for making complex bar charts for business plus fancy greeting cards and geometric designs. Great for homework too. **Everyone must have a 4 color printer plotter for their VIC-20 or Commodore-64.** List $199.00. **Sale $99.00.**

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY • BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

cardco, inc.

It's time for your computer to grow up.

VIC-20 EXPANDER SALE!

	LIST	SALE
• 8K RAM MEMORY EXPANDER (Free $16.95 Game)	$ 69.95	**$39.00**
• 16K RAM MEMORY EXPANDER "CARDCO" (Free $29.95 Adventure Game)	$ 99.00	**$49.00**
• 24K RAM MEMORY EXPANDER (Free $29.95 Adventure Game)	$159.00	**$59.00**
• 32K RAM MEMORY EXPANDER (Expands to total memory to 57K (57,000 bytes)	$199.00	**$75.00**
• 3 SLOT "CARDCO" SWITCHABLE EXPANDER BOARD	$ 39.95	**$29.00**
• 6 SLOT "CARDCO" SWITCHABLE "CARDCO" RIBBON CABLE EXPANDER BOARD (Lifetime warranty)	$ 99.00	**$49.00**
• 40-80 COLUMN BOARD	$ 99.00	**$59.00**
• "CARDCO" PARALLEL PRINTER INTERFACE	$ 99.00	**$69.00**

VIC-20 WORD PROCESSOR SALE

1. **"Write Now" Word Processor Cartridge!** (Better than Quick Brown Fox) No disk drive or cassette player needed! Just plug in the "Write Now" cartridge. Easy to use and learn, has all the professional features: margin settings, editing word wrap, scrolling, search and replace, centering, page numbering, user defined characters, ascii code set that allows use of all printer features!! Includes a powerful mailmerge for mailing lists! List $39.95 **Sale $34.95.**

2. **"H.E.S. Writer" Word Processor Cartridge!** Full screen editing, scrolling up and down, word wrap, left and right justification, centering, page numbering, screen text preview and more! List $39.95. **Sale $24.95.**

3. **"Total Text" Word Processor creates professional documents!** Features include page numbering, right and left margins, paragraph control, upper/lower case, full screen editing, footnotes, etc.! (8K Ram required) Tape List $34.95. **Sale $19.95.**

4. **"Total Labels" Mailing List!** Keep and print mailing list labels with ease. Add change and delete labels, alphabetize, option to select any part of list. (8K Ram required) Tape List $24.95. **Sale $14.95.**

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master Card. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

VIC 20

40-80 COLUMN BOARD

only $59⁰⁰

SALE SALE

Now you can get 40 or 80 Columns on your T.V. or monitor at one time! No more running out of line space for programming and making columns. Just plug in this board and you immediately convert your VIC-20 computer to 40 or 80 columns! Comes in an attractive molded case with instructions! List $99.00. **Sale $59.00.**

FOR ONLY $24.95 you can get a 40-80 Column Board "WORD PROCESSOR" with mail merge and terminal emulator PLUS! AN ELECTRONIC SPREAD SHEET (like Visicalc) the word processor requires 8K—mail merge 16K! List $59.00. **Sale $39.90.** *If purchased with board only $24.95. (Tape or Disk.)

WE LOVE OUR CUSTOMERS!

COMMODORE 64

80 COLUMN BOARD

$99⁰⁰

SALE

Now you can program 80 columns on the screen at one time! Converts your Commodore 64 to 80 columns when you plug in the PROTECTO 80 Expansion Board. List $199.00. **Sale $99.00.**

FOR ONLY $24.95 you can get an 80 Column Board "WORD PROCESSOR" with mail merge and terminal emulator PLUS! AN ELECTRONIC SPREAD SHEET (like Visicalc) List $59.00. **Sale $39.90.** *If purchased with board only $24.95. (Tape or Disk.)

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SANYO MONITOR SALE!!

9" Data Monitor

- 80 Columns × 24 lines
- Green text display
- Easy to read - no eye strain
- Up front brightness control
- High resolution graphics
- Quick start - no preheating
- Regulated power supply
- Attractive metal cabinet
- UL and FCC approved

- *15 Day Free Trial - 90 Day Immediate Replacement Warranty*

9" Screen - Green Text Display	$ 69.00
12" Screen - Green Text Display (anti-reflective screen)	$ 99.00
12" Screen - Amber Text Display (anti-reflective screen)	$ 99.00
12" Screen-Super 1000 Line Amber Text Display	$129.00
14" Screen - Color Monitor (national brand)	$249.00

Display Monitors From Sanyo

With the need for computing power growing every day, Sanyo has stepped in to meet the demand with a whole new line of low cost, high quality data monitors. Designed for commercial and personal computer use. All models come with an array of features, including up-front brightness and contrast controls. The capacity 5 × 7 dot characters as the input is 24 lines of characters with up to 80 characters per line.

Equally important, all are built with Sanyo's commitment to technological excellence. In the world of Audio/Video, Sanyo is synonymous with reliability and performance. And Sanyo quality is reflected in our reputation. Unlike some suppliers, Sanyo designs, manufactures and tests virtually all the parts that go into our products, from cameras to stereos. That's an assurance not everybody can give you!

SANYO

Official Video Products of the Los Angeles 1984 Olympics

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

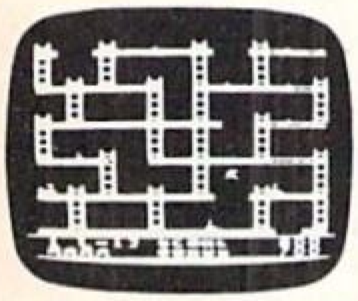
PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

*Buy 1 (at Sale Price!) Get One FREE

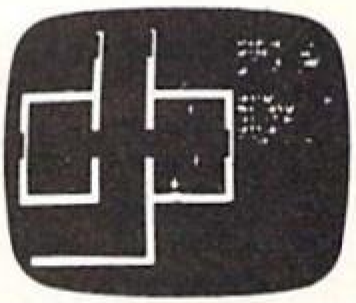
Commodore 64

EPYX

Award Winning Games

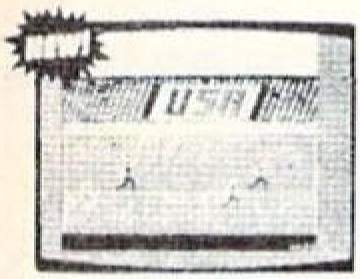


Jumpman — If you like Donkey Kong, you'll love Jumpman. Over 30 different screens with 8 speeds and 5 skill levels make this the fastest action game in the country. You must leap girders, climb ropes, and scale ladders to reach and diffuse bombs while avoiding robots, birds, bullets, explosives, crumbling walls, vanishing escape routes, and many other obstacles. (Truly a fantastic game!) List $39.95. **Sale $24.95** (Disk/Tape.)

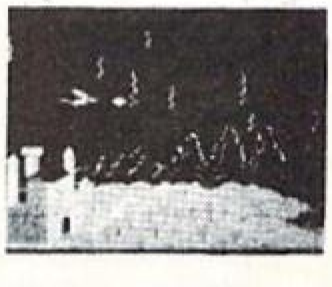


Temple of Apshai — ***(computer game of the year)***. This is the standard by which other adventure games are judged. Full color graphics portray the temple and all its contents — magic, monsters, doomed cities and damsels in distress. Do battle in real time with over 20 monsters, expansion modules will keep your adventure alive in the future. List $39.95. **Sale $24.95**. (tape/disk).

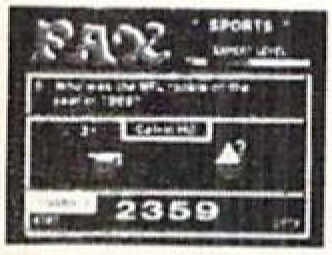
(Also Available For VIC-20)



Summer Games — Now you can compete in the Olympics right along with the real competitors. Compete in track, swimming, diving, cycling, archery, weight lifting, rowing and more. This Olympic game is so realistic it even includes a beautiful opening ceremony and an awards presentation after each event. (Highly recommended, by far the best Olympics game we've seen). List $39.95. **Sale $28.00**. (disk only).



Dragon Riders of Pern — Based on the best selling book series by Anne McCaffey (over 10 million sold). This program forces you to make an alliance with the Lord holders, then mount your dragon (stunning graphics) and battle the dreaded thread. (Super fast action with a great theme). List $39.95. **Sale $24.00**. (tape/disk).



Fax — Stop wasting quarters! Now you can play Exidy's Fax at home. Play against the computer or an opponent to get the answer first, over 3,700 questions dealing with sports, television, movies, histories, and many more subjects will keep you playing for hours (a must for trivia buffs). List $34.95. **Sale $24.95**. (disk only).

* Buy Any Epyx Game From Protecto And Send The Proof Of Purchase Seals To Epyx And Epyx Will Send You A Free Game.

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master-Card. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

AT LAST!!! A REAL FLIGHT SIMULATOR!!!

FOR THE COMMODORE 64 COMPUTER!

ONLY $29.95 Cassette or Disk

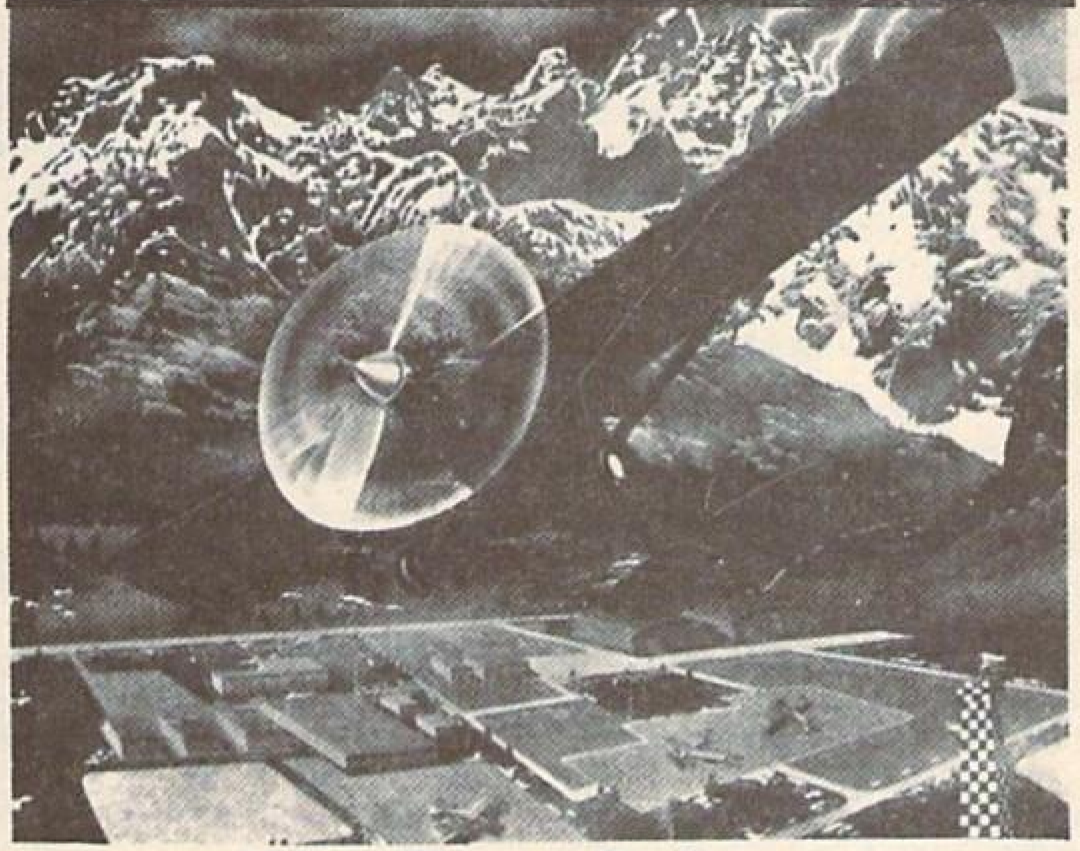
list $39.95

From The Creators of HELLCAT ACE!

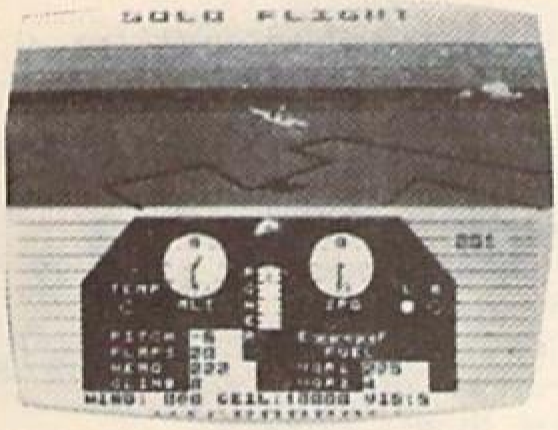
A real flight simulator allows you, the pilot, to accomplish Takeoffs, Landings, Cross-Country Navigation, and Emergency Procedure practice under Clear (VFR), Cloudy (IFR), and Cross-Wind conditions. SOLO FLIGHT is such an advanced simulation that uses realistic three dimensional terrain graphics, actual configuration instrument panel, multiple air navigation maps, and accurate aircraft performance characteristics to provide the challenge, thrill, excitement and joy of the flight experience!!!

- Three Dimensional Terrain Graphics
- Multiple Airports and Runways
- Dual Radio Navigation Instruments
- Full Feature, Actual Configuration Instrument Panel
- Multiple Air Navigation Charts
- Instrument Approaches

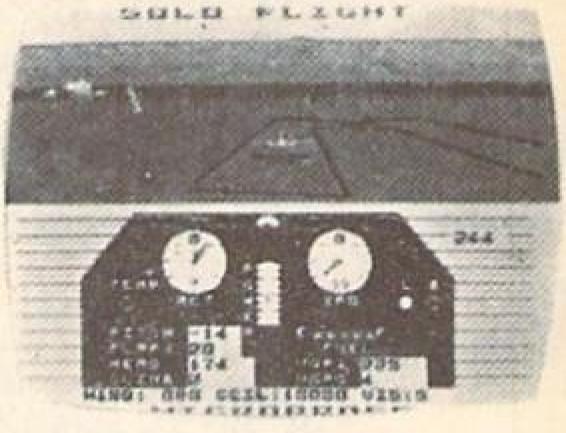
- Takeoffs
- Landings
- Cross-Country Navigation — Day, VFR and IFR Flying
- Emergency Procedures Practice
- Air Mail Delivery Game for 1-4 Players
- Smooth Landing & Good Navigation Scoring



MicroProse Software



SOLO FLIGHT



Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Commodore 64
No. 1
Educational Software
At Its Best

SPINNAKER™
We make learning fun.

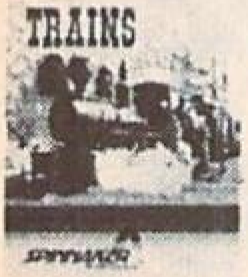
Sale!!

KIDS ON KEYS
SPINNAKER

Kids on Keys — Letters, numbers, and pictures rain down the screen. It's up to you to find the matching keys or type in the correct word before the images disappear. Helps familiarize the children with the computer keyboard as well as identifying numbers, letters, and words. List $34.95. **Sale $22.00.** **(Cartridge/Disk)**

IN SEARCH OF THE MOST AMAZING THING
SPINNAKER

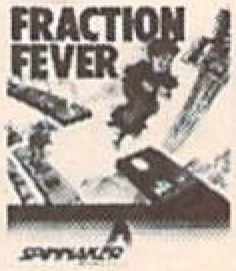
In Search of the Most Amazing Thing — You have been given a quest to find the most amazing thing. You must negotiate with the aliens as well as use your problem solving skills to overcome obstacles along the way. Fantastic for learning step by step thinking to overcome problems. Highly recommended!!
List $39.95. **Sale $28.00.** **(Disk)**



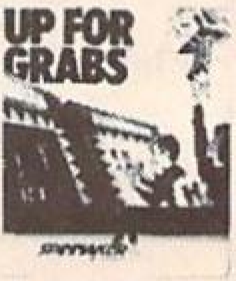
Trains — Enter the world of Big Business. Keep the railroad going and growing. Teaches fundamentals of financial management (and planning). This is a must for any youngster and most adults. Very highly recommended.
List $39.95. **Sale $28.00** **(Disk)**

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

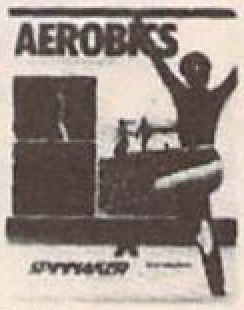
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master Card. We ship C.O.D.



Fraction Fever — This fast paced arcade graphic program helps you develop: better understanding of what a fraction is, of relationships between different fractions, and numerical and visual representation of fractions. To succeed you must match the right fractions, zap the wrong ones, look out for holes in the floor, and watch the clock. (Fast paced, highly recommended!!).
List $34.95. **Sale $22.00.** **(Cartridge)**



Up For Grabs — If you like Scrabble you'll love this one. You have to grab letters off a rotating cube before anyone else and then use the letters to make words in a hurry. Fantastic for spelling and vocabulary skills.
List $39.95. **Sale $28.00.** **(Cartridge)**



Aerobics — Follow the instructions on the screen to fun and fitness. Including warm ups, aerobics, stretches, and cool-downs. Fantastic graphics let you follow the characters exactly. Tailors your exercise program to your needs. Best fitness program we've seen. List $44.95. **Sale $36.00.** **(Disk)**

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

COMMODORE 64 SUPER

Fantastic Sale!

HES Special Programs For Your Computer

While They Last!! **Call Today!!** **First Come Basis!!**

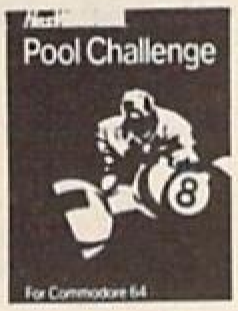
SUPER ZAXXON Sale $27.95

• Fantastic Graphics & Sound • Best Arcade Game • Super Action

The smash hit arcade game is now a reality on your Commodore-64. This advanced version of Zaxxon includes multi-level gun emplacements, tunnels and even the fire breathing dragon. (Highly recommended). List $39.95. **Sale $27.95.**

HesWare
Zaxxon
For Commodore 64
SEGA

List $39.95
Sale $27.95



List $29.95
Sale $24.95

POOL CHALLENGE Sale $24.95

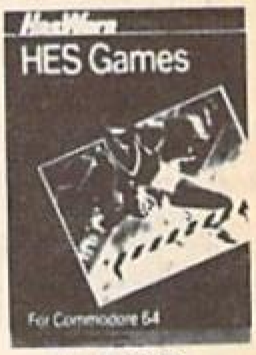
• 6 Games in One • Endorsed by Minnesota Fats • Great Action

Now you can play pool on your Commodore-64. Sharpen your aiming accuracy and shooting finesse. Using only a joystick and a few keystrokes you choose where to aim and how much power to use. Fantastic action. (Cartridge). List $29.95. **Sale $24.95.**

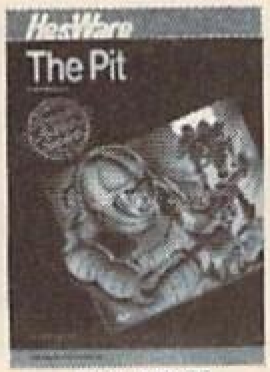
HES GAMES Sale $29.95

• 6 World Class Events • Fantastic Realism • Instant Replay

Now you can have the Olympics in your home. Compete in 6 events including weightlifting, springboard diving, hurdles, 100 meter dash, long jump and archery. This one will keep you playing for hours. (Disk). List $39.95. **Sale $29.95.**



List $39.95
Sale $29.95



List $29.95
Sale $22.95

THE PIT Sale $22.95

• Arcade Classic • Fantastic Graphics • Great Sound

This arcade classic has all the realism of the arcade with tunneling minors, rampant robots, missiles, tanks and even a pit monster. This game will keep you playing for hours. (Cartridge). List $29.95. **Sale $22.95.**

We Have All HES Products At Super Sale Prices!!

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

EDUCATIONAL PROGRAM SALE!!

VIC 20

$19.95

was $34.95

Educational Software

SPINNAKER™

We make learning fun.

No. 1 At Its Best

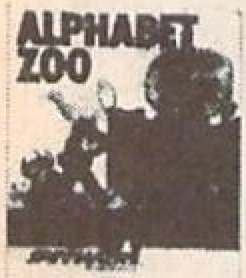
KINDER COMP

Kindercomp—is for ages 3 through 8. Tasks include matching shapes and letters, writing names, drawing pictures and filling in missing numbers. KINDERCOMP teaches reading readiness and counting skills. (Cartridge). List $34.95. **Sale $19.95.**

STORY MACHINE

Story Machine—is a computerized storybook come to life. Children enter their own story which the computer animates with full color graphics and sound. STORY MACHINE is for ages 5 to 9 and teaches sentence structure and keyboard recognition. (Cartridge). List $34.95. **Sale $19.95.**

Face Maker—lets children create funny faces that can be animated to wink, smile or wiggle their ears. FACEMAKER teaches computer fundamentals such as menus, cursors, simple programs and graphics to ages 4 to 12. (Cartridge). List $34.95. **Sale $19.95.**



Alphabet Zoo—helps children strengthen their letter recognition skills as they associate letters of the alphabet with the sounds they represent. The game also helps sharpen a child's spelling skills. Alphabet Zoo is for ages 3 to 8. (Cartridge). List $34.95. **Sale $19.95.**

KIDS ON KEYS

Kids on Keys—familiarizes children with the computer keyboard as they learn to identify numbers, letters, and words, and associate them with images they see on the screen. Kids on Keys is for ages 3 to 9. (Cartridge). List $34.95. **Sale $19.95.**

Now developing a child's mind can be fun for the whole family.

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master-Card. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Factory Clearance

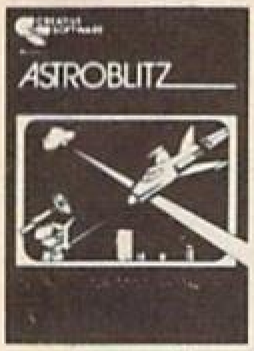
VIC-20 Cartridge Games

CREATIVE SOFTWARE

No. 1 Seller In U.S.A.

Sale

First Come Basis



Astroblitz — (Like Defender.) Voted No. 1 computer game overall by Creative Computing! Fantastic copy of an arcade classic. Destroy alien saucers and gun towers while avoiding enemy objects that float by. List $34.95. **Sale $7.95.**

Your Choice

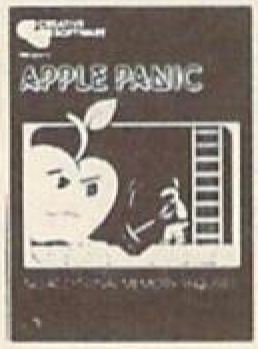
$7.95

Was $34.95

CREATIVE SOFTWARE
SERPENTINE

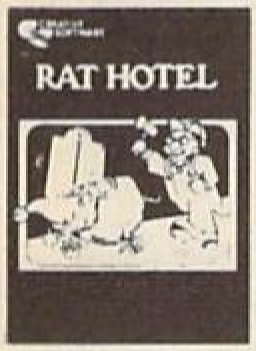
Serpentine — Awarding winning game. Avoid hostile red snakes and lay your young while getting food to keep you alive. Fantastic strategy game that has become a classic. List $34.95. **Sale $7.95.**

Priced Below Cost

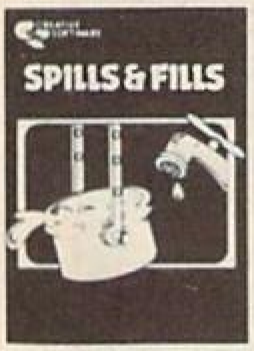


Apple Panic — The fast action computer game hit now comes to the VIC-20. You must dig traps in the brick floor and wait for the wandering apples to fall in. Then simply beat them on the head to push it through and destroy the apple. Otherwise the apple's will roll right over you and kill you. (Fast action.) List $34.95. **Sale $7.95.**

While They Last

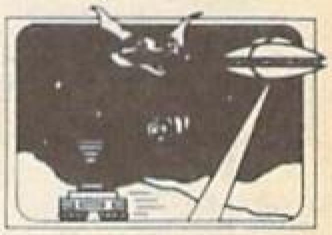


Rat Hotel — Here you get to play the rat. Running and jumping from floor to floor in Hotel Paridisimo you must eat whatever you find but Waldo the maintenance man has set many traps for you that you must avoid. Plus Waldo has an atomic elevator that let's him jump several floors at once. See if you can get to the bottom before Waldo does you in. Fantastic graphics and sound. List $34.95. **Sale $9.95.**



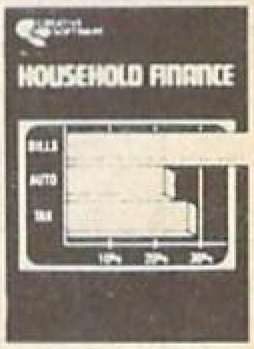
Spills & Fills — How much is too much? How much isn't enough? With this original concept educational game designed for grade schoolers through high school students, you can develop ratio and perspective skills. With three exciting and challenging skill levels, this program presents new and ever-changing challenges for the inquisitive mind. **Sale $9.95.**

Trashman (Better than Pacman). Dash for the trash and pick up CASH!!! List $29.95. **Sale $9.95.**



Terraguard — Your mission is to destroy the aliens before their ship has a chance to land. Move your base along the ground and shoot up at them. List $34.95. **Sale $9.95.**

Video Mania — You're in an open field and strange alien creatures are roaming all over. If you stand still too long, they'll collide with you, causing instant annihilation. Your only defense is to destroy them by throwing your electronic, football-shaped alien zapper at them. **Sale $9.95.**



Household Finances — This 4 part program will take care of all your family's finances. Record, change and delete expense and income items from 16 categories, monthly and yearly totals are easily graphed and even a budgeting program to help you clamp down on excessive spending. List $34.95. **Sale $9.95.**

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master-Card. We ship C.O.D.

Shutter Bug *(Continued From Page 32)*

time spent in the maze, the number of shots taken and the last shot you took (the negative to the right of the view finder). Also, when your camera is wound, an arrow on the screen indicates this.

The game is written completely in BASIC (even the 3-D view), and the same program runs on both the 64 and VIC 20 (it tests location zero to see which computer it is running on). The 64 version is really the VIC 20 version centered and minus the disappearing map. The program is extremely compact in order to allow it to run on the unexpanded VIC, so be very careful when typing it in.

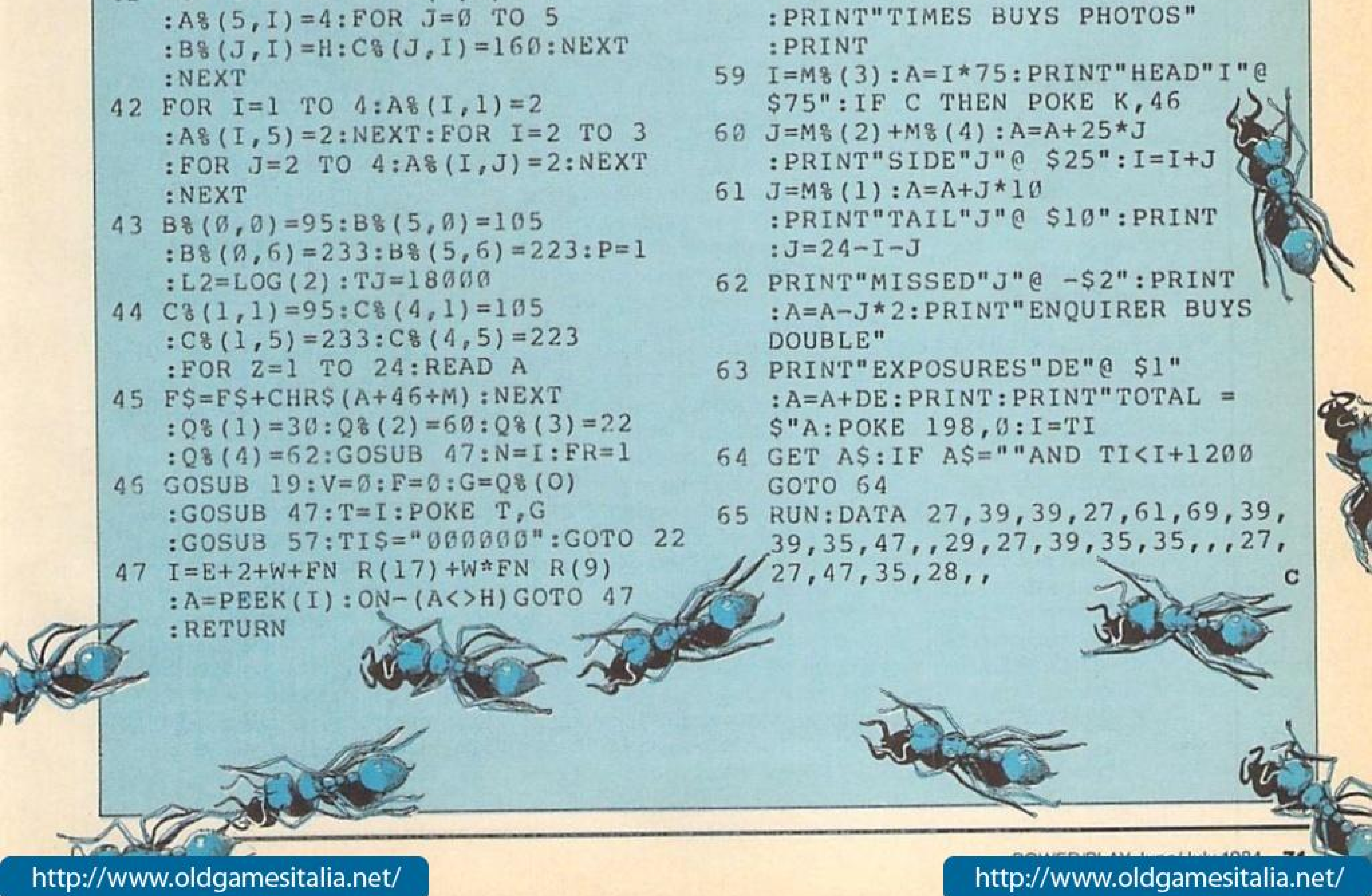
The maze-drawing algorithm works with any size maze, though it does not have a guaranteed completion time. The 3-D view is kept in three two-dimensional arrays, and is optimized to prevent redrawing doors that are already open or "undrawing" ones that are not there.

The VIC 20 joystick is re-encoded to look like the 64's joystick. This is then decoded using the LOG() function, which is a slow function (though faster than a single 2↑X) but only requires a single call, decoding all the bits in the same amount of time. In the case of two switches down at the same time, LOG() will give you the higher of the two. Logarithms of base two are used, LOG(x)/LOG(2).

Shutter Bug

```
 1 DIM A%(5,6),B%(5,6),C%(5,6),
   O%(5),Q%(4),M%(4):I=RND(-TI)
   :GOTO 24
 2 PRINT C$"[DOWN]"INT(TI/60)
   :IF TI>TJ GOTO 58
 3 Q=-2*(PEEK(T+O%(O))<M)-4*(PEEK
   (T+O%(O-1))<M)-(PEEK(T+O%(O+1))
   <M)
 4 FOR I=0 TO 5:X=I+S:FOR J=0 TO 6
   :A=A%(I,J):IF A=0 GOTO 8
 5 IF J=0 THEN IF P THEN IF(A AND
   Q)=(A AND V)THEN J=6:GOTO 8
 6 Z=W*J+X:IF A AND Q THEN POKE Z,
   C%(I,J):GOTO 8
 7 POKE Z,B%(I,J)
 8 NEXT:NEXT:V=Q:RETURN
 9 A=PEEK(D):IF C THEN A=(A/4 AND
   7)+(8 AND PEEK(D+1)/16)+(A/2
   AND 16)
10 IF(A AND 31)=31 THEN RETURN
11 ON LOG(NOT A AND 31)/L2 GOTO
   14,15,17,48:I=T+O%(O)
12 IF PEEK(I)=H THEN POKE T,H:T=I
   :POKE T,G
13 RETURN
14 I=T-O%(O):GOTO 12
15 O=1-O*(O<4)
16 G=Q%(O):POKE T,G:RETURN
17 O=O-1-4*(O=1):GOTO 16
18 I=N+O%(L):IF PEEK(I)=H THEN
   POKE N,H:N=I:GOTO 20
19 I=L:L=1+FN R(4):IF I=L GOTO 19
20 POKE N,Q%(L):RETURN
21 GOSUB 9:GET A$:IF A$<>""THEN
   PRINT C$"[DOWN,SPACE2,SHFT I]"
   :PRINT B$" [BACK ARROW,SHFT K]
   ":WD=1
22 GOSUB 2:GOSUB 18
   :IF T+O%(O)<>N THEN ON R+1
   GOTO 21:R=0:P=0:GOSUB 3:P=1
   :GOTO 21
23 R=1:B=L-O-4*(O>L):FOR Y=0 TO 2
   :Z=U+W*Y:FOR X=0 TO 1
   :POKE X+Z,FN F(B):NEXT:NEXT
   :GOTO 21
24 DEF FN R(X)=INT(RND(1)*X)
   :M=128:DEF FN F(Z)=ASC(MID$
   (F$,1+Z*6+Y*2+X))
25 IF PEEK(0)=76 THEN C=1:W=22
   :L=7680:D=37151:POKE D+3,127
   :K=36867:GOTO 27
26 W=40:L=1024:D=56320
   :POKE 53281,1
27 FOR I=1 TO 18:D$=D$+" "
   :C$=C$+"[SHIFT *]":NEXT
   :A$=LEFT$(D$,W/2-6)
   :B$=LEFT$(A$,W/2-10)
28 PRINT"[CLEAR]":PRINT A$"
   [BLACK,SHFT U,SHIFT *8,SHFT I]
   ":PRINT A$"[SHFT -,CYAN,
   CMDR *,RVS,SPACE6,RVOFF,
   SHFT POUND,BLACK,SHFT -]"
   :PRINT A$"[SHFT -] [CYAN,
   CMDR *,RVS,SPACE4,RVOFF,
   SHFT POUND,BLACK] [SHFT -]"
29 PRINT A$"[SHFT -,SPACE2,CYAN,
   SPACE4,BLACK,SPACE2,SHFT -,
   CMDR Q,SHFT +,SHIFT *2,SHFT +,
   CMDR W]":FOR I=1 TO 3
   :PRINT A$"[SHFT -,SPACE8,
   SHFT -,CMDR Q,CMDR W,SPACE2,
   CMDR Q,CMDR W]":NEXT
30 PRINT A$"[SHFT -,SPACE2,GREEN,
   SPACE4,BLACK,SPACE2,SHFT -,
   CMDR Q,SHFT +,SHIFT *2,SHFT +,
   CMDR W]":PRINT A$"[SHFT -]
   [GREEN,RVS,SHFT POUND,SPACE4,
   CMDR *,RVOFF,BLACK] [SHFT -]"
31 PRINT A$"[SHFT -,GREEN,RVS,
   SHFT POUND,SPACE6,CMDR *,
   RVOFF,BLACK,SHFT -]"
   :PRINT A$"[SHFT J,SHIFT *8,
   SHFT K]":PRINT B$"[RVS,SHFT U]
   "C$"[SHFT I]"
32 FOR I=1 TO 9:PRINT B$"[RVS,
   SHFT -,BLACK]"D$"[SHFT -]"
   :NEXT:PRINT B$"[RVS,SHFT J]
   "C$"[SHFT K,RVOFF]";
33 E=L+12*W+W/2-11:S=L+3*W+W/2-4
   :O=1:H=160:GOSUB 47:H=32
   :U=S+2+2*W
34 R%=S+10+2*W:O%(1)=-W:O%(2)=-1
   :O%(3)=W:O%(4)=1:O%(0)=1
   :O%(5)=-W
35 Q%(1)=-W-1:Q%(2)=W-1:Q%(3)=W+1
   :Q%(4)=1-W:C$="[HOME]"+B$
36 POKE I,H:Z=6:FOR Y=FN R(3)TO 5
   :A=I+O%(Y):IF PEEK(A)<>160
   GOTO 39
37 F=0:G=0:J=0:FOR X=1 TO 4
   :F=F-(PEEK(A+Q%(X))=H)
   :IF PEEK(A+O%(X))=H THEN G=G+1
   :J=X-J
38 NEXT:IF F<4 AND G<2 OR(G<3
   AND F<1)OR(G=2 AND
   ABS(J)=2)THEN Z=Y:Y=6
39 NEXT:IF Z<6 THEN I=I+O%(Z)
   :V=V+1:GOTO 36
40 IF V<95 THEN GOSUB 47:GOTO 36
41 FOR I=0 TO 6:A%(0,I)=1
   :A%(5,I)=4:FOR J=0 TO 5
   :B%(J,I)=H:C%(J,I)=160:NEXT
   :NEXT
42 FOR I=1 TO 4:A%(I,1)=2
   :A%(I,5)=2:NEXT:FOR I=2 TO 3
   :FOR J=2 TO 4:A%(I,J)=2:NEXT
   :NEXT
43 B%(0,0)=95:B%(5,0)=105
   :B%(0,6)=233:B%(5,6)=223:P=1
   :L2=LOG(2):TJ=18000
44 C%(1,1)=95:C%(4,1)=105
   :C%(1,5)=233:C%(4,5)=223
   :FOR Z=1 TO 24:READ A
45 F$=F$+CHR$(A+46+M):NEXT
   :Q%(1)=30:Q%(2)=60:Q%(3)=22
   :Q%(4)=62:GOSUB 47:N=I:FR=1
46 GOSUB 19:V=0:F=0:G=Q%(O)
   :GOSUB 47:T=I:POKE T,G
   :GOSUB 57:TI$="000000":GOTO 22
47 I=E+2+W+FN R(17)+W*FN R(9)
   :A=PEEK(I):ON-(A<>H)GOTO 47
   :RETURN
48 ON WD+1 GOTO 54:IF T+O%(O)<>N
   THEN A$="NOTHING THERE":F=0
   :I=H:J=0:GOTO 55
49 A$="CLICK":B=L-O-4*(O>L)+1:F=B
   :M%(B)=M%(B)+1
50 FOR Y=0 TO 2:Z=R%+W*Y
   :FOR X=0 TO 1:POKE X+Z,
   FN F(B-1)-M:NEXT:NEXT
51 IF MS THEN MS=0:POKE K,46
   :GOTO 56
52 IF C AND FN R(9)=1 THEN MS=1
   :A$="ANT TOOK MAP":POKE K,24
   :GOTO 56
53 POKE N,H:GOSUB 47:N=I:GOTO 56
54 A$="DOUBLE EXPOSURE"
   :FR=FR+1*(FR>0):DE=DE+1
   :M%(F)=M%(F)-1:F=0:I=66:J=16
55 FOR Y=0 TO 2:Z=R%+W*Y
   :FOR X=0 TO 1:POKE X+Z,
   I+FN R(J):NEXT:NEXT
56 PRINT C$SPC(10-LEN(A$)/2)A$
   :FOR I=1 TO 500:NEXT:WD=0
57 PRINT C$D$:PRINT:PRINT
   :PRINT B$FR:PRINT B$"[SPACE3]"
   :PRINT B$"[SPACE3]":FR=FR+1
   :IF FR<25 THEN RETURN
58 PRINT"[CLEAR]GAME OVER":PRINT
   :PRINT"TIMES BUYS PHOTOS"
   :PRINT
59 I=M%(3):A=I*75:PRINT"HEAD"I"@
   $75":IF C THEN POKE K,46
60 J=M%(2)+M%(4):A=A+25*J
   :PRINT"SIDE"J"@ $25":I=I+J
61 J=M%(1):A=A+J*10
   :PRINT"TAIL"J"@ $10":PRINT
   :J=24-I-J
62 PRINT"MISSED"J"@ -$2":PRINT
   :A=A-J*2:PRINT"ENQUIRER BUYS
   DOUBLE"
63 PRINT"EXPOSURES"DE"@ $1"
   :A=A+DE:PRINT:PRINT"TOTAL =
   $"A:POKE 198,0:I=TI
64 GET A$:IF A$=""AND TI<I+1200
   GOTO 64
65 RUN:DATA 27,39,39,27,61,69,39,
   39,35,47,,29,27,39,35,35,,,27,
   27,47,35,28,,
```

C



Hangman in LOGO

by David Malmberg
Fremont, California

Let's continue our quixotic exploration of the LOGO computer language for the Commodore 64. In previous columns we had an overview of LOGO's turtle graphics and list processing capabilities, developed an interactive game called LOGO Lander, and discussed books and other resources that will help you to learn about LOGO. In this column, we will put much of what we have already learned together to develop another game, LOGO Hangman. This program will use both the turtle graphics and the list-manipulation facilities of LOGO. By studying the listing as you key in the program you will learn a great deal about the language and how its procedural structure can be used to adapt simple games and graphics to the Commodore 64.

The game of Hangman is normally played with paper and pencil by two people. One person picks a secret word and the other person tries to guess it. Initially blank spaces are shown for each letter. For example, you might start the game with:

___ ___ ___ ___ ___ ___ ___ ___ ___,

indicating that the word is nine letters long. You guess one letter at a time. If you guess a letter that is in the secret word, that letter is substituted for the blank where ever the letter occurs in the word. For example, after a few guesses the word might be shown as: ___ cie ___ ti ___ t. At this point you should recognize the word as "scientist" and by announcing that fact win this round of the game.

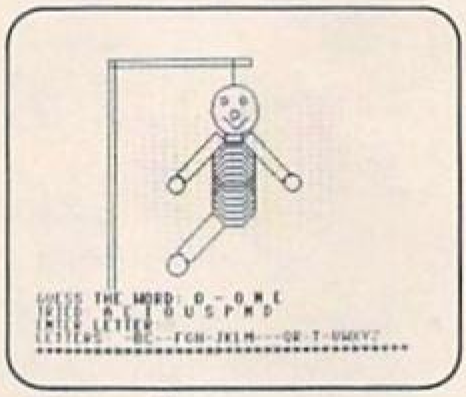


Figure 1.

Whenever you guess a letter that is not in the word, the other person draws part of a body hanging in a gallows. The first "missed" letter causes a head to be drawn. The second wrong guess draws the neck. After seven wrong guesses all of the body parts will have been drawn and you lose that round of the game.

In the LOGO version of Hangman, the Commodore 64 is your opponent and picks its words from a diabolical vocabulary. If you do not like the program's list of words, you may substitute your own words. Using lists of spelling words will make this program a painless (and even fun) way to improve spelling grades. As you enter your letter guesses, the program will display the alphabet and a separate list of the letters you have tried in the bottom part of the screen. Also shown will be the blanks of the secret word with your correct letter guesses in their proper position. When you guess wrong, LOGO's turtle graphics capabilities are used to draw the body parts on the top of the screen. See Figure 1. for an example of how the screen will look about half-way through a typical game.

A few words of explanation on each of the procedures will make following the logic of the program easier.

The first procedure in the listing, Hangman, produces an animated title page. The command RUN :BODY.PARTS draws each body part in turn. Calling the FROWN procedure changes the body's smile to a frown. The procedure then asks you to press any key. Once you press a key, the GAME procedure is called and play begins.

The second procedure in the listing, INITIALIZE, performs a number of start-up functions, including: (1) making the list of letters already tried (:TRIED) empty, (2) setting up the complete list of :BODY.PARTS, (3) executing the PICK.WORD procedure, (4) clearing and splitting the screen so graphics will be drawn on the top twenty lines and text will be written on the bottom five lines and finally (5) drawing the gallows.

PICK.WORD first makes sure you will get random numbers when you need them. Then it sets up the game's :VOCABULARY of secret words. If you want to change the vocabulary, just substitute your own list. Next PICK.WORD calls the procedure RANDOM.LIST to make a random selection from the :VOCABULARY list and makes it the secret :WORD. Next the number of letters in :WORD is counted and a list of blank characters of the same length is created. Finally, :ALPHABET is initialized.

GET.LETTER first prints the blank characters and the correct letter guesses in their proper position. Then it gets your next :LETTER guess. If the :LETTER is one you have already tried, an error message is printed and you start the whole procedure again. If :LETTER is a new guess, it is first added to the list :TRIED. Then :LETTER is tested to see if it is a member of the set of characters in the secret :WORD. If it is, the call to the procedure SWAP changes the blank character(s) to the :LETTER in their proper position(s). Finally, the reference to DELETE causes the :LETTER to be replaced by a blank in the :ALPHABET.

The TRY procedure in conjunction with GET.LETTER is the heart of the program. First, TRY calls GET.LETTER. Then it tests if you have guessed all of the letters in the secret word by testing if there are any blank characters left. If you have guessed the word correctly, the procedure tells you how many tries it took and asks if you want to play again. If there are still blank characters (and hence letters to guess), TRY tests if the current :LETTER is in :WORD. If this test results in a true condition, TRY is executed recursively.

If your :LETTER guess is not in :WORD, then the command RUN SENTENCE FIRST :BODY.PARTS [] causes the next body part to be drawn. Then that body part is deleted from the list of :BODY.PARTS. If that was the last element of :BODY.PART then all the parts have been drawn and you lost the round; the smile is erased and a frown drawn on the face; the secret word is revealed and you are asked if you wish to play again. If that was not the last element of :BODY. PARTS, the game continues by TRYing again.

The procedures GALLOWS, BOX, POLYGON, HEAD, NECK, TRUNK, RIGHT.ARM, LEFT.ARM, RIGHT.LEG and LEFT.LEG are all straight forward turtle graphics routines to draw the shapes their names describe.

David Malmberg

LOGO Hangman

```
TO HANGMAN
    INITIALIZE
    CURSOR 13 20 PRINT [LOGO HANGMAN]
    CURSOR 10 21 PRINT [BY DAVID MALMBERG]
    CURSOR 0 23 PRINT [GUESS THE SECRET WORD BEFORE YOU HANG!]
    RUN :BODY.PARTS
    FROWN
    CURSOR 10 24 PRINT1 [PRESS ANY KEY]
    MAKE "LETTER READCHARACTER
    GAME
END


TO INITIALIZE
    MAKE "TRIED []
    MAKE "BODY.PARTS [HEAD NECK TRUNK RIGHT.ARM LEFT.ARM
          RIGHT.LEG LEFT.LEG]
    PICK.WORD
    NODRAW
    CLEARSCREEN
    SPLITSCREEN
    BACKGROUND 14
    PENCOLOR 6
    HIDETURTLE
    GALLOWS
END


TO PICK.WORD
    RANDOMIZE
    MAKE "VOCABULARY [BYTE FUCHSIA ALLOCATE QUEUE ADOBE
            BOOKKEEPER BASIC APRON TUREEN SWAN VETERINARY
            GHOST GIRAFFE ANTARCTIC MAGNET APROPOS PYRAMID
            OZONE COMPUTER PIZZA LOGO KANGAROO FEIGN UGLY
            SLEEPER MOUSTACHE SCIENTIST RETURN MAIDEN QUININE]
    MAKE "WORD ( RANDOM.LIST :VOCABULARY )
    MAKE "LENGTH ( COUNT :WORD )
    MAKE "BLANK []
    REPEAT :LENGTH [MAKE "BLANK LPUT "- :BLANK]
    MAKE "ALPHABET "ABCDEFGHIJKLMNOPQRSTUVWXYZ
END


TO RANDOM.LIST :LIST
    OUTPUT ITEM ( 1 + RANDOM ( COUNT :LIST ) ) :LIST
END


TO GAME
    INITIALIZE
    TRY
END


TO TRY
    GET.LETTER
    IF NOT MEMBER? "- :BLANK THEN CURSOR 0 23 PRINT1
          SENTENCE [YOU GOT IT ON THE] COUNT :TRIED
          PRINT [-TH TRY!!] AGAIN?

    TEST MEMBER? :LETTER :WORD
    IFTRUE TRY
    RUN SENTENCE FIRST :BODY.PARTS []
    MAKE "BODY.PARTS BUTFIRST :BODY.PARTS
    IF :BODY.PARTS = [] THEN FROWN CURSOR 0 22 PRINT
          SENTENCE [THE WORD WAS:] :WORD
          PRINT [YOU LOST AND MUST HANG!!] AGAIN?
    TRY
END
```

```
TO GET.LETTER
    CURSOR 0 20 PRINT SENTENCE [GUESS THE WORD:] :BLANK
    CURSOR 0 22 PRINT1 [ENTER LETTER:]
    MAKE "LETTER READCHARACTER
    CURSOR 0 24 PRINT1 [***************************************]
    TEST MEMBER? :LETTER :TRIED
    IFTRUE CURSOR 10 24 PRINT1 SENTENCE [YOU ALREADY TRIED]
          :LETTER GET.LETTER
    MAKE "TRIED LPUT :LETTER :TRIED
    CURSOR 0 21 PRINT SENTENCE [TRIED:] :TRIED
    TEST MEMBER? :LETTER :WORD
    IFTRUE MAKE "BLANK SWAP :LETTER :BLANK :WORD
          CURSOR 16 20 PRINT :BLANK
    MAKE "ALPHABET DELETE :LETTER :ALPHABET
    CURSOR 0 23 PRINT SENTENCE [LETTERS:] :ALPHABET
END


TO AGAIN?
    CURSOR 12 24 PRINT1 [PLAY AGAIN ?]
    MAKE "LETTER READCHARACTER
    TEST :LETTER = "Y
    IFTRUE GAME
    IFFALSE NODRAW PRINT [THANKS FOR THE GAME!!] TOPLEVEL
END


TO SWAP :L :B :W
    IF EMPTY? :W THEN OUTPUT []
    TEST :L = ( FIRST :W )
    IFTRUE OUTPUT FPUT :L SWAP :L ( BUTFIRST :B ) ( BUTFIRST :W)
    IFFALSE OUTPUT FPUT ( FIRST :B ) SWAP :L ( BUTFIRST :B)
          ( BUTFIRST :W )
END


TO DELETE :L :Q
    IF :Q = " THEN OUTPUT "
    TEST :L = FIRST :Q
    IFTRUE OUTPUT WORD "- DELETE :L ( BUTFIRST :Q )
    IFFALSE OUTPUT WORD ( FIRST :Q ) DELETE :L ( BUTFIRST :Q )
END


TO GALLOWS
    PENUP
    SETXY ( - 100 ) ( - 100 )
    SETHEADING 0
    PENDOWN
    BOX 200 5
    PENUP
    SETXY ( - 100 ) 100
    SETHEADING 90
    PENDOWN
    BOX 115 5
    PENUP
    SETXY 0 100
    SETHEADING 180
    PENDOWN
    FORWARD 20
END


TO BOX :A :B
    REPEAT 2 [FORWARD :A RIGHT 90 FORWARD :B RIGHT 90]
END
```

(Continued On Next Page)

```
TO HEAD
    SETXY 0 80
    SETHEADING 95
    POLYGON 20 6
    PENUP
    SETXY 0 60
    PENDOWN
    POLYGON 10 2
    PENUP
    SETXY ( - 8 ) 70
    PENDOWN
    POLYGON 15 1
    PENUP
    SETXY 8 70
    PENDOWN
    POLYGON 15 1
    PENUP
    SETXY ( - 8 ) 55
    SETHEADING 135
    PENDOWN
    FORWARD 14
    LEFT 90
    FORWARD 14
    PENUP
END

TO POLYGON :N :S
    REPEAT :N [FORWARD :S RIGHT 360/:N]
END

TO NECK
    SETXY ( -7 ) 42
    SETHEADING 90
    PENDOWN
    BOX 14 7
END

TO TRUNK
    SETHEADING 90
    SETXY ( -7 ) 37
    REPEAT 7 [POLYGON 8 14 PENUP SETHEADING 180
          FORWARD 5 SETHEADING 90 PENDOWN]
END

TO RIGHT.ARM
    PENUP
    SETXY ( -7 ) 37
    SETHEADING 225
    PENDOWN
    FORWARD 45
    RIGHT 90
    FORWARD 5
    RIGHT 180
    POLYGON 20 2
    RIGHT 180
    FORWARD 5
    RIGHT 90
    FORWARD 45
    SETXY ( -7 ) 37
    PENUP
END

TO LEFT.ARM
    PENUP
    SETXY 7 37
    SETHEADING 135
    PENDOWN
    FORWARD 45
    LEFT 90
    FORWARD 5
    POLYGON 20 2
    FORWARD 5
    LEFT 90
    FORWARD 45
    SETXY 7 37
    PENUP
END

TO RIGHT.LEG
    PENUP
    SETXY ( -7 ) ( - 27 )
    SETHEADING 225
    PENDOWN
    FORWARD 40
    RIGHT 90
    FORWARD 7
    RIGHT 180
    POLYGON 25 2
    RIGHT 180
    FORWARD 7
    RIGHT 90
    FORWARD 40
    SETXY ( -7 ) ( - 27 )
    PENUP
END

TO LEFT.LEG
    PENUP
    SETXY 7 ( - 27 )
    SETHEADING 135
    PENDOWN
    FORWARD 40
    LEFT 90
    FORWARD 7
    POLYGON 25 2
    FORWARD 7
    LEFT 90
    FORWARD 40
    SETXY 7 ( - 27 )
    PENUP
END

TO FROWN
    SETXY ( -8 ) 55
    SETHEADING 135
    PENDOWN
    PENERASE
    FORWARD 14
    LEFT 90
    FORWARD 14
    PENUP
    PENCOLOR 6
    SETXY ( -8 ) 48
    SETHEADING 75
    PENDOWN
    FORWARD 10
    RIGHT 30
    FORWARD 10
END
```

C

PEEK
FULL FEATURE MAGAZINE
on
CASSETTE
FOR THE VIC 20™ and COMMODORE 64™

- GAMES
- EDUCATION
- REVIEWS
- TUTORIALS
- BUSINESS
- UTILITIES

• READY TO RUN PROGRAMS •

1 YEAR (12 issues) . . $49.95   6 MOS. (6 issues) $28.95

PLEASE SPECIFY VIC 20™ or COMMODORE 64™
8K Expander required for Vic 20™

subscribe today
PEEK MAGAZINE
P.O. BOX 35167
CLEVELAND, OHIO 44135

VIC 20™ and COMMODORE 64™ is a trademark of Commodore Electronics Ltd.

poke into peek
MAGAZINE

Commodore 64 Sprite Contest Winners

by Betsy Byrne
Associate Editor

I received *bundles* of entries for the sprite contest announced in the Winter, 1983, issue of *Power/Play*. In fact, the mailman on my route has been distinctly cool towards me lately. I was receiving entries right up to deadline day, and it was tough choosing the winners.

An interesting phenomenon occurred, however—all my entries were in age category C (age 12-18). Since we had originally planned to have a winner in each of *three* age groups, I was stumped—but only for a minute. "Why not just have a first, second and third prize instead!" I exclaimed to myself. So that's exactly what I did.

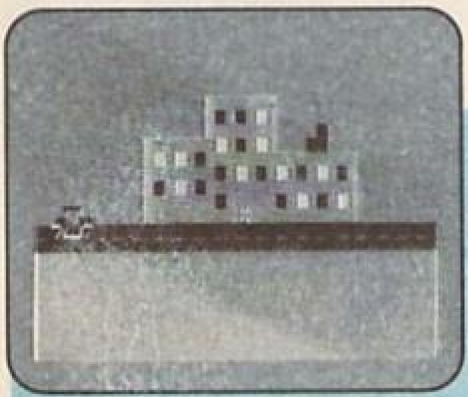
It was a bit hard choosing the first place winner, since I had three entries that were running neck and neck. Fortunately, all three sprites were entered by one person so I never did choose, I just gave them *all* first place.

The winner is....... John Dolinar, age 18, from Sagamore Hills, Ohio. His three sprites—Antique Car, Guitar and Spriteman—all showed imagination, talent and craftsmanship. As all of us know who have experimented with sprite creation, it's not as easy as it looks! If you type in John's Antique Car sprite you may be surprised at what you see when you run it.

Our second place winner is Bonnie Baker, age 13, from Weston, Massachusetts. Bonnie's Sailboat sprite came to me in the form of a game: the Sailboat sprite is manuevered around a group of buoys and tries to get as close to the buoys as possible, without crashing of course! Bonnie had some questions about making the boat sink to the bottom of the screen when it hits a buoy, and having it make a scraping sound as it hits. I gave her questions to Tim Villanueva, my friendly neighborhood sprite expert, and he promises some interesting answers for the next issue of *Power/Play*.

Third place was captured by Mitch Shaw, 14, from Pensacola, Florida. He called his sprite Sprite Three, but I named it "Tigger the Tiger", and it was love at first sight. I mean, this thing is *cute*, folks!

So there you have it, our first annual sprite-making contest. Did I say "First Annual?" Does that mean there will be another contest *this* year? You will just have to wait and see—but be sure to SAVE all your most spritacular efforts, and—don't take any wooden sprites!

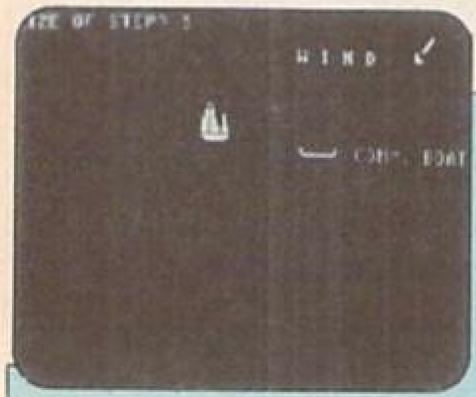


Antique Car

```
0 REM ***************************
  ***
1 REM * ANTIQUE CAR
  : JOHN DOLINAR *
2 REM ***************************
  ***
3 PRINT"[CLEAR]": POKE 53280,6
  :POKE 53281,6
4 V=53248:POKE V+21,3:POKE V+39,2
  :POKE V+40,2:POKE V+28,3
  :POKE V+37,12:POKE V+38,0
6 POKE 2040,13:POKE 2041,14
8 POKE V+23,3:REM POKEV+29,3
10 FOR ZZ=832 TO 894:READ Z
   :POKE ZZ,Z:NEXT
12 FOR YY=896 TO 958:READ Y
   :POKE YY,Y:NEXT
14 POKE 53248,00:POKE 53249,160
16 POKE 53250,23:POKE 53251,160
20 PRINT"[DOWN4]"
21 PRINT"[RIGHT17,ORANGE,RVS,
   SPACE7,RIGHT4,RVOFF,GRAY1,
   SHFT I,SHFT U,RIGHT,SHFT U]"
22 PRINT"[RIGHT17,ORANGE,RVS]
```

(Continued On Next Page)

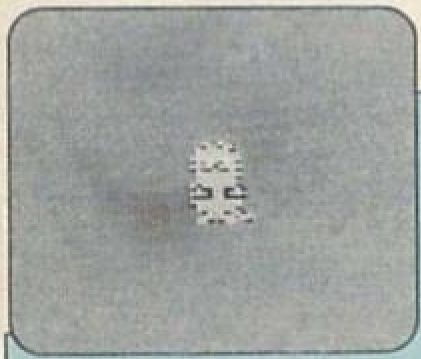
```
   [BLACK] [ORANGE] [BLACK]
   [ORANGE] [YELLOW] [ORANGE]
   [RVOFF,GRAY1,RIGHT4,SHFT U2]"
23 PRINT"[RIGHT17,RVS,ORANGE,
   SPACE7,RIGHT4,BLACK] "
24 PRINT"[RIGHT11,ORANGE,RVS,
   SPACE7,YELLOW] [ORANGE]
   [BLACK] [ORANGE] [YELLOW]
   [ORANGE] [RIGHT3,BLACK,SPACE2,
   RVOFF]"
25 PRINT"[RIGHT11,RVS,ORANGE]
   [YELLOW] [ORANGE] [YELLOW]
   [ORANGE] [BLACK] [ORANGE,
   SPACE15,RVOFF]"
26 PRINT"[RIGHT11,RVS,ORANGE,
   SPACE7,BLACK] [ORANGE]
   [YELLOW] [ORANGE] [BLACK]
   [ORANGE] [YELLOW] [ORANGE]
   [BLACK] [ORANGE] [YELLOW]
   [ORANGE] [BLACK] [ORANGE]
   [RVOFF]"
27 PRINT"[RIGHT11,RVS,ORANGE]
   [BLACK] [ORANGE] [YELLOW]
   [ORANGE] [BLACK] [ORANGE,
   SPACE15,RVOFF]"
28 PRINT"[RIGHT11,RVS,ORANGE,
   SPACE7,BLACK] [ORANGE,SPACE5,
   BLACK] [ORANGE] [YELLOW]
   [ORANGE] [BLACK] [ORANGE]
   [YELLOW] [ORANGE] [RVOFF]"
29 PRINT"[RIGHT11,RVS,ORANGE,
   SPACE9,RVOFF,SHFT Z,RVS,
   SPACE11]"
30 FOR F=1 TO 40:PRINT"[GRAY1,
   RVS] [RVOFF]";:NEXT
31 FOR F=1 TO 20:PRINT"[GRAY1,
   RVS,CMDR T] [RVOFF]";:NEXT
32 FOR F=1 TO 40:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
33 FOR F=1 TO 40:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
34 FOR F=1 TO 40:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
35 FOR F=1 TO 40:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
36 FOR F=1 TO 40:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
37 FOR F=1 TO 40:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
38 FOR F=1 TO 40:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
39 FOR F=1 TO 39:PRINT"[GREEN,
   RVS] [RVOFF]";:NEXT
40 POKE 2023,160:POKE 2023+54272,5
41 PRINT"[HOME]";
42 FOR I=0 TO 232
43 POKE 53248,I:POKE 53249,160
44 POKE 53250,I+23:POKE 53251,160
45 NEXT
46 POKE V+16,2
47 FOR I=1 TO 23
48 POKE 53248,I+232
   :POKE 53249,160
49 POKE 53250,I:POKE 53251,160
50 NEXT
51 POKE V+16,3
52 FOR I=23 TO 111
53 POKE 53248,I-23:POKE 53249,160
54 POKE 53250,I:POKE 53251,160
55 NEXT
56 POKE 53264,0:GOTO 42
99 END
100 DATA 0,63,255,0,48,0,0,40,0,
    15,250,0,63,250,128,85,250,
    128,63,122,170
200 DATA 253,223,255,247,213,85,
    63,192,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0
300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
400 DATA 0,0,0,128,0,0,128,16,0,
    191,240,0,190,176,0,245,80,0,
    223,240,0
500 DATA 95,112,0,93,240,0,15,
    240,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0
600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
```



Sailboat

```
0 PRINT"[CLEAR,DOWN8]"
1 REM MOVABLE MARIO
2 PRINT"WHEN SIZE OF STEP SHOWS
  ON SCREEN":PRINT
3 PRINT"TYPE A NUMBER FROM 1 AND
  10.":PRINT:PRINT:PRINT
4 PRINT"WHEN SAILBOAT SHOWS ON
  SCREEN":PRINT"PRESS U TO MOVE
  UP"
5 PRINT"H TO MOVE LEFT"
6 PRINT":J TO MOVE RIGHT":PRINT
```

```
  :PRINT"AND N TO MOVE DOWN"
7 PRINT:PRINT"TRY TO MOVE THE
  SAILBOAT SPRITE AROUND THE
  NUMBERED BUOYS.YOU START"
8 PRINT"IN BETWEEN THE COMM.
  BOAT AND BUOY #1.[SPACE2]
  END BETWEEN BOAT AND #2.";
9 PRINT"GET AS CLOSE TO THE
  BUOYS AS POSSIBLE W/OUT
  HITTING THEM.TIME YOURSELF";
10 PRINT" AND TRY TO BEAT THAT
   TIME NEXT GAME."
16 FOR X=0 TO 200:NEXT
   :PRINT"[CLEAR]"
17 POKE 53281,0:POKE 53280,1
18 PRINT"SIZE OF STEP";:INPUT Z
19 V=53248:POKE V+21,1
   :POKE V+39,2:POKE V+28,1
   :POKE V+37,6:POKE V+38,1
20 POKE 2040,13
21 D=100:H=150
22 FOR N=0 TO 62:READ Q
   :POKE 832+N,Q:NEXT
23 PRINT"[SPACE35,YELLOW,SHFT N]"
24 PRINT"[SPACE24,YELLOW]W I N D
   [SPACE3,CMDR Z]"
25 D=100:H=150
26 PRINT:PRINT:PRINT:PRINT
   :PRINT"[SPACE19,RVS,RED]1";
27 PRINT:PRINT"[SPACE24,YELLOW,
   SHFT J,SHFT C2,SHFT K] [GREEN]
   COMM. BOAT"
28 PRINT:PRINT:PRINT:PRINT
   :PRINT"[SPACE29,RVS,RED]3"
29 PRINT:PRINT:PRINT:PRINT
   :PRINT"[SPACE10,RVS,RED]2"
30 IF D<0 THEN D=255
31 IF D>255 THEN D=0
32 IF H<0 THEN H=255
33 IF H>255 THEN H=0
35 POKE V+0,H:POKE V+1,D
39 POKE V+0,H:POKE V+1, D
50 GET A$:IF A$=""THEN 50
55 IF A$="H"THEN H=H-Z
60 IF A$="J"THEN H=H+Z
70 IF A$="U"THEN D=D-Z
80 IF A$="N"THEN D=D+Z
90 IF A$="Q"THEN PRINT"[CLEAR]"
   :POKE V+21,0:STOP
99 GOTO 30
100 DATA 0,0,0,1,128,0,3,128,0,6,
    128,0,6,192,0,14,192,0,14,
    192,128
200 DATA 30,224,192,30,224,192,
    62,240,192,62,240,224,126,
    248,224,126,248,224
300 DATA 126,252,240,126,252,240,
    126,254,248,66,128,128,127,
    255,248,31,255,240
310 DATA 15,255,224,0,0,0
```



Tiger Sprite

```
0 PRINT"
  [CLEAR,DOWN]
  POWER/PLAY
  SPRITE
  CONTEST"
1 PRINT"ENTRY
  #3 BY MITCH SHAW [DOWN5]"
2 PRINT"WHEN SIZE OF STEP SHOWS
  ON SCREEN":PRINT
3 PRINT"TYPE A NUMBER FROM 1 TO
  INFINITY.":PRINT:PRINT"TYPE Q
  TO QUIT":PRINT
4 PRINT"WHEN OBJECT SHOWS ON
  SCREEN":PRINT"PRESS U TO MOVE UP"
5 PRINT"H TO MOVE LEFT"
  :PRINT"J TO MOVE RIGHT"
  :PRINT"AND N TO MOVE DOWN"
6 PRINT"[HOME,DOWN20]
  PRESS ANY KEY":GET Z$
  :IF Z$=""THEN 6
7 PRINT"[CLEAR]":POKE 53281,6
  :POKE 53280,0
8 PRINT"[WHITE]SIZE OF STEP";
  :INPUT Z
10 V=53248:POKE V+21,1
   :POKE V+39,8:POKE V+28,1
   :POKE V+37,0:POKE V+38,1
20 POKE 2040,13
22 PRINT"[DOWN]EXPAND SPRITE";
   :INPUT A$:IF LEFT$(A$,
   1)="Y"THEN POKE V+23,1
   :POKE V+29,1
23 PRINT"[CLEAR]"
24 IF LEFT$(A$,1)<>"Y"THEN POKE
   V+23,PEEK(V+23)AND 254
   :POKE V+29,PEEK(V+29)AND 254
25 D=100:H=200
26 REM IT'S GARFIELD
27 FOR N=0 TO 62:READ Q
   :POKE 832+N,Q:NEXT
30 IF D<0 THEN D=255
31 IF D>255 THEN D=0
32 IF H<0 THEN H=255
33 IF H>255 THEN H=0
```

(Continued On Page 81)

Backwords!

For the VIC 20 and Datassette

by Dr. Petrie Curryfavor
Komputerist and Professor of Obfuscation

*(**Ed. Note:** We found this letter stuffed under the front door one day in March. Since Betsy Byrne claims the good Doctor is sane, we agreed to run yet another of his mysterious code programs. Once again, we take no responsibility for spelling, punctuation, capitalization, etc.)*

Hi there, Ms. LeBold and Ms. Le Byrne!

I Know yoU have been eagerly awaiting the NEXT Communication from Dr. Curryfavor, and Here it Is! Brilliant old Dr. Curryfavor comes through Again with *another* Mysterious and *baffling* Code Book! He was going to give it to the C.i.A. to replace their stodgy *old* Codes that everyBody knows already, but when he saw the wONdErful backwardS letters program by Karen Pierce in your winTer issue, he decided that *you* should have it instead. Obviously, you have More of the proper Spirit! Great Minds running on the same Track!

Of course, cunning old Dr. Curryfavor's program is very different from Ms. Pierce's program; hers PRINTs the LETTERS backwards, while "BACKWORDS!" PRINTs the WORDS and *senTences* backwards! Too bad there's not Enough RooM in an *UN*-eXpanded VIC to hold *both* of the programs at once. Hmmm....

Where Was I? Oh, yes! "BACKWORDS!" Now the reason this program is so BIG is beCause it does a lot—you can not only type *backwardS* in Two differenT ways, you can also PRINT backwards on your printer (if you have one), SAVE your MESSage on tape, DECODE the message, and a few Other Things.

On tOp of *that,* crafty old Dr. C. has added a nifty little AniMated title page. Let's lOOk at that first:

100-180

Dr. Curryfavor has Heard that there is SomeThing called "VIC 20 proGramming." yoU may have noticed that there is Not much memory in your Un-eXpanded VIC—so we VIC 20 ProGrammers Must "crUNch" our programs to fit. The routines in these lines give some examples of "Curryfavor Crunches" (No, it is NOT a BreakFast cereal!)

lOOk at line 100 and the DATA statements in Lines 610 and 620—see how Three lines can Do the work of Seven? This can save Quite a Bit (no Pun UNintended) of memory. We are also using *One* animation Routine to manipulate *two* $tring$. MID$(BW$,22-B,1) will PRINT *one* letter from BW$, and 22-B is the number of the letter it will PRINT. There are six blank spaces at the Front of BW$ and B goes from 6 TO 15, so 22-B starts counting from 16 TO 7. TAB(B) Starts the PRINTing from the *Left* (columns 6 through 15.) See how 22-B gets SMAller when B gets bigGER? Now, at the Same Time, right after we PRINT MID$(BW$,22-B,1), we PRINT MID$ (BW$,7,15-B). This means that we Start PRINTing BW$ Beginning at letter 7 ("B" or "!") and PRINT one less letter each time (15-B takes the letter oFF oF the righT enD oF thE worD. Lines 150 and 160 change the variables to keep the animation going (did you know that $-1*-1=1$? And $1*-1=-1$?) IF you hit any key, line 170 will send you out of the LooP, but you may have to wait a few seconds!

170

Where it sends you, is to the "Vacuum Cleaner" in line 170! This routine eMpTys out the inPut buffer so that the key you hit doesn't PRINT in line 240.

200

Here is wHere we CLeaR out our variables, and DIMension our aRrAy. This is a two DIMensional aRrAy which will hold data for 19 lines of Up to 23 characters (number 23 is for a carriage RETURN, if necessary.)

210-240

This is the Main Menu. No Mysteries here!

250-280

IF your choice was "1", then we "Default" to the values in line 250. IF "2" was your choice, we change those to the values in line 260. If you lOOk at line 330, and study the results of using the different $tring$, Helpful old Dr. Curryfavor thinks you will see whY the "SECRET" mode only PRINTs One letter at a time in the UPper LEft corner. IF I=3 THEN we GO TO the "UTILITIES MENU."

290-380

This is the Main LooP, where you eNter your MESSage. Line 300 has the counters for the two DIMension$ in our array—IF there are more than 22 keystrokes, we add a line (A=A+1) and re-set B to the First Letter of the new line (lOOk at line 350). Be sure to hit RETURN when you get near the end of a line (just like a typeWriter), or the print-Out will lOOk strange. IF there are more than 18 lines, THEN line 310 sends us to the "FILE FILLED" warning, so plan your MESSage carefully. 320 is our "ESCape Clause"; Any time you want to GO TO the "UTILITIES MENU", hit the left arrow key.

Line 340 is our DELete routine. Yes, the DELete key *does* work in this program! Do you see how it sub-tracts from the array? Line 350 builds the array. You can think of it as a Grid, just like the screen display—with rows (A) and columns (B). IF you press RETURN, THEN in 360 we start a new line (BeGinning in the Left column, and Adding a new line.) THEN we GO TO the "DISPLAY" SUBroutine...

390-410

... which is right here! lOOk at how the array is DISPLAYed backwardS! We start with the hiGHest numbers of both rows and columns!

430

This line waits for you to hit the left arrow key, which will let you fall through to the...

440-500

... UTILITIES MENU! Line 470 allows you to start a new MESSage, but gives you a chance to change your Mind! Line 480 allows you to CONTINUE your entry (does line 280 make sense now?) 490 handles the jobs of fiVe IF... THEN statements! CRUNCH! CRUNCH!

510

This ONE line is our DECODEr! An array makes things very comPACT!

520-530

Here's our "SAVE" routine... and Here's our array again!

540-580

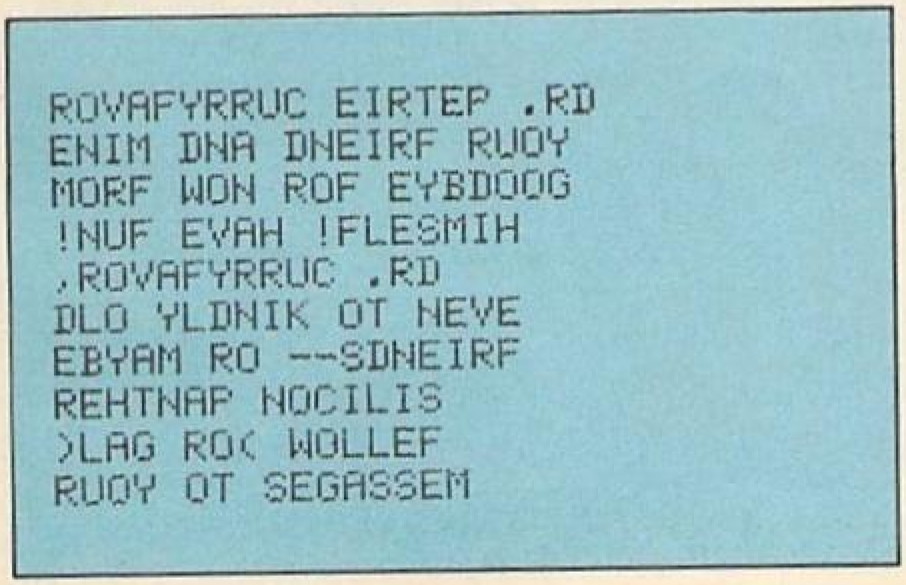
Our "LOAD" lOOks a lot like our "SAVE." Notice how we set "A" for our DISPLAY routine (line 580.) Be Sure you have your tape positioned correctly for these two routines!

590-600

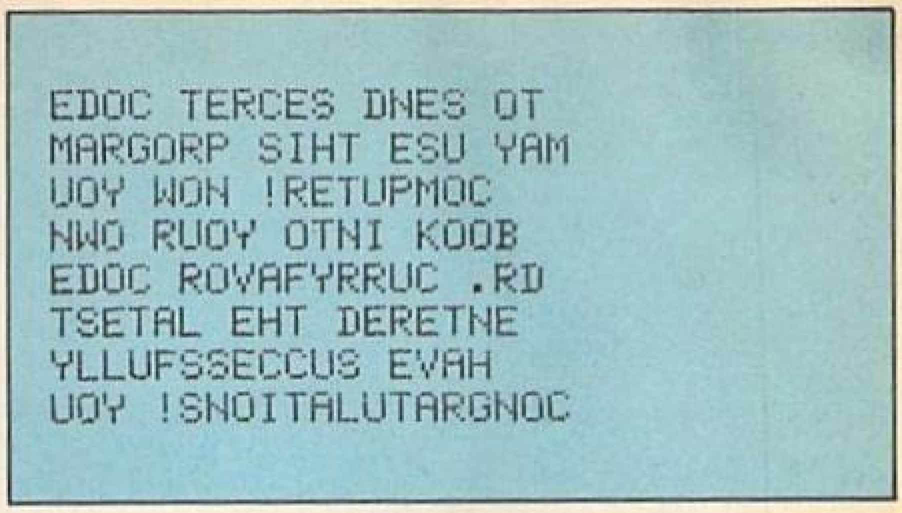
I hope you have a Printer hooked up! OtherWise, you will get a "DEVICE NOT PRESENT ERROR" IF you select "3" from the UTILITIES MENU. If you don't have a printer yet, you had better leave these lines out for Now, and type: 590 RETURN. This will keep your program from "bombing" until you get your printer.

The print-out is 22 columns wide, of course—just the right size to be cut out and hidden in a chewing gum wrapper! A Perfect way to smuggle MESSages to other Silicon Panther Youth Society members (S.P.Y.S., for short!) Naturally, you could also pass on a DaTassette tape, instead.

NOW, to be Sure your TranslatOR is working, type in this MESSage from Friendly old Dr. Curryfavor:



```
ROVAFYRRUC EIRTEP .RD
ENIM DNA DNEIRF RUOY
MORF WON ROF EYBDOOG
!NUF EVAH !FLESMIH
,ROVAFYRRUC .RD
DLO YLDNIK OT NEVE
EBYAM RO --SDNEIRF
REHTNAP NOCILIS
)LAG RO( WOLLEF
RUOY OT SEGASSEM
```



```
EDOC TERCES DNES OT
MARGORP SIHT ESU YAM
UOY WON !RETUPMOC
NWO RUOY OTNI KOOB
EDOC ROVAFYRRUC .RD
TSETAL EHT DERETNE
YLLUFSSECCUS EVAH
UOY !SNOITALUTARGNOC
```

Now, some people complain that unAppreciated old Dr. Curryfavor Never does any programs for the Commodore 64. Well, Dr. Curryfavor doesn't *have* a 64 yet, but he *was* able to borrow one for a while, so he could Tell you how to translate the program for it.

To BeGin with, the 64 has More Memory and Only 80-column proGram lines, so some of the longer lines need to be Broken into two lines. Line 210 should be re-Written as *Two* PRINT statements, divided between "**** BACKWORDS! **** " and "**** MAIN MENU **** ". Then, at the beginning of the program, you must add this line:

```
10 POKE 53281,1:CL=54272
```

to turn the screen White. CL is the difference between the CoLor meMORy map and the screen memory map.

Then, since the 64 has a 40-column screen, instead of the 22 columns on the VIC, we should do something about esTABlishing a 22-column MargiN, so that our MES$age will be "transportable" to your fellow S.P.Y.s' VIC 20's. *One* way to do that is like this:

```
285 FOR M=1046 TO 2006 STEP40:POKE M,101:POKE
M+CL,2:NEXT
```

which will draw a vertIcal Red line on the screen that will be the enD of your page. Be careful not to type past it! You could also add a MarGin Bell, like in "IGPAY" from the SprinG issue of *Power/Play.* I'll bet yoU could even figure out an EDiting routine and a Routine that would preVent you from tyPing more than 22 CHaRacters. Try it!

If you have aNy problems with this program, Please let amiable old Dr. Curryfavor kNOw about them. And, if you would Rather Not type the program in yourSelf, send $3.00 with a blank Tape or Disk and a S.A.S.E. to:

Dr. PeTrie Curryfavor
c/o Eddie Johnson
2928 Tennessee NE
Albuquerque, NM 87110

Checks or Money Orders should be made out to Eddie Johnson. Dr. Curryfavor is mad at the Bank and Won't go in there any more!

Well, that's It for This issue, but Keep in touch—coming Up soon aRe some *more* Secret Code Books from the CryptoGraphic ReSearch Labs of your friend and Mine,

Dr. Petrie Curryfavor,
Cryptomaniac and
Professor of Obfuscation

Listing on next page

```
1 REM ***** BACKWORDS! *****
2 REM BY DR. CURRYFAVOR, 1984
3 REM 2928 TENNESSEE NE,
  ALBUQUERQUE, NM 87110  505)
  299-1662
4 REM DO NOT TYPE IN REM
  STATEMENTS OR PROGRAM WILL *
  NOT * RUN!!
100 PRINT"[CLEAR]";:FOR P=1 TO 7
    :READ S:READ D$
    :PRINT TAB(S)D$:NEXT
110 BW$="[SPACE6]BACKWORDS!
    [SPACE6]":Z=-1
120 PRINT"[HOME,RVS,MAGENTA]"BW$
    :FOR T=1 TO 999:NEXT
130 FOR B=6 TO 15:PRINT"[HOME,
    BLUE,RVOFF]"TAB(B)MID$(BW$,
    22-B,1)"[MAGENTA,RVS]
    "MID$(BW$,7,15-B)
140 FOR T=1 TO 99:NEXT:NEXT
    :FOR T=1 TO 999:NEXT
150 Z=Z*-1
160 IF Z=1 THEN BW$="[SPACE6]
    !SDROWKCAB[SPACE6]":GOTO 120
170 GET B$:IF B$<>""THEN 190
180 GOTO 110
190 FOR V=1 TO 10:GET V$:NEXT
200 CLR:DIM W$(19,23):A=1
210 PRINT"[CLEAR,RVS]
     **** BACKWORDS! ****[SPACE2]
    **** MAIN[SPACE2]MENU **** "
220 PRINT"[DOWN] SELECT NUMBER"
    :PRINT"[DOWN] AND PRESS [RVS,
    RED] RETURN [RVOFF,BLUE]"
230 PRINT"[DOWN] (1) SECRET
    ENTRY":PRINT"[DOWN]
     (2) VISIBLE ENTRY"
    :PRINT"[DOWN] (3) UTILITIES
    [DOWN]"
240 INPUT I:IF I<1 OR I>3 THEN
    210
250 S$="[MAGENTA,RVS,HOME]"
    :R$="[BLUE,RVOFF] "
260 IF I=2 THEN S$="[MAGENTA,RVS]
    ":R$="[BLUE,RVOFF]"
270 IF I=3 THEN 440
280 PRINT"[CLEAR]";
    :IF J=7 THEN GOSUB 390
290 GET A$:IF A$=""THEN 290
300 B=B+1:IF B>22 THEN B=1:A=A+1
310 IF A>18 THEN 420
320 IF A$="[BACK ARROW]"THEN 440
330 PRINT S$A$R$;
340 IF A$=CHR$(20)THEN W$(A,
    B-1)="":A$=""
350 W$(A,B)=A$
360 IF A$=CHR$(13)THEN B=0:A=A+1
    :GOSUB 390
370 IF A$=CHR$(32)THEN GOSUB 390
380 GOTO 290
390 PRINT"[CLEAR]":FOR C=A TO 1
    STEP-1:FOR D=22 TO 1 STEP-1
400 PRINT W$(C,D);
410 NEXT:NEXT:PRINT"[HOME]";
    :RETURN
420 PRINT"[DOWN,RED,RVS,LEFT]
     FILE FILLED [BLUE,RVOFF]"
430 PRINT"[HOME,RVS,MAGENTA]
     * PRESS [BACK ARROW]
     FOR MENU * [BLUE]";:GET K$
    :IF K$<>"[BACK ARROW]
    "THEN 430
440 PRINT"[CLEAR,RVS]
     BACKWORDS UTILITIES: "
450 PRINT"[DOWN] (1) SAVE"
    :PRINT"[DOWN] (2) LOAD"
    :PRINT"[DOWN] (3) PRINT"
    :PRINT"[DOWN] (4) DISPLAY"
460 PRINT"[DOWN] (5) DECODE"
    :PRINT"[DOWN] (6) DESTROY"
    :PRINT"[DOWN] (7) CONTINUE
    [DOWN]":INPUT J
470 IF J=6 THEN INPUT"[CLEAR,RED,
    RVS] ZAP MEMORY [BLUE,RVOFF]
    ";J$:IF J$="Y"THEN 200
    :GOTO 440
480 IF J=7 THEN 210
490 ON J GOSUB 520,540,590,390,
    510:GOTO 430
500 GOTO 440
510 PRINT"[CLEAR]":FOR C=1 TO 19
    :FOR D=1 TO 23:PRINT W$(C,D);
    :NEXT:NEXT:RETURN
520 INPUT"[CLEAR]SAVE";F$
```

```
    :OPEN 1,1,2,F$
530 FOR C=1 TO A:FOR D=1 TO 22
    :PRINT#1,W$(C,D);:NEXT:NEXT
    :CLOSE 1:RETURN
540 INPUT"[CLEAR]LOAD";F$
    :OPEN 1,1,0,F$
550 FOR C=1 TO 19:FOR D=1 TO 22
    :GET#1,G$:W$(C,D)=W$(C,D)+G$
560 IF G$=""THEN 580
570 NEXT:NEXT
580 CLOSE 1:A=C+1:RETURN
590 OPEN 1,4:CMD 1:FOR C=19 TO 1
    STEP-1:FOR D=23 TO 1 STEP-1
    :PRINT#1,W$(C,D);
600 NEXT:NEXT:PRINT#1:CLOSE 1
    :RETURN
610 DATA 94,"CODE[SPACE2]BOOK",4,
    "[DOWN]ENCRYPTION BY
    :",0,"[DOWN]DR. PETRIE
    CURRYFAVOR,"
620 DATA 3,"REVERSIONIST AND",5,
    "[DOWN]PROFESSOR OF",4,"
    [DOWN]RECIPROCATION.",71,
    "HIT ANY KEY"
```

Typelefter

Editors:

I was afraid of this! You *did* have to go and encourage Dr. Curryfavor by publishing one of his "un-word processors" in the Spring issue and now look what happens! I thought I had safely hidden my VIC 20, but he found it anyway. At any rate, this latest exercise in absurdity was again addressed to you, and was left to me to handle the mailing. Doesn't it seem to you that a published author ought to provide his own postage, at least?

I decided after looking at Dr. Curryfavor's message on my printer, that I would write my own decoder program for it. Thus, TYPELEFTER—a program simple enough for even an adult to type in! After all, I have much more serious things to do than to copy in a big, complicated program like BACKWORDS!

Of course, TYPELEFTER doesn't have all the features that Dr. Curryfavor's BACKWORDS! has, but it does have a "backwards carriage return" (lines 120-130), and will translate any backwards writing into a sensible form. It starts printing from the bottom right corner of the screen and moves left as you type, and up one line to the right hand column if you hit RETURN.

In closing, I should probably warn you that the indefatigable Dr. Curryfavor is threatening to send more code games to you in the near future. You have my sympathies. **C**

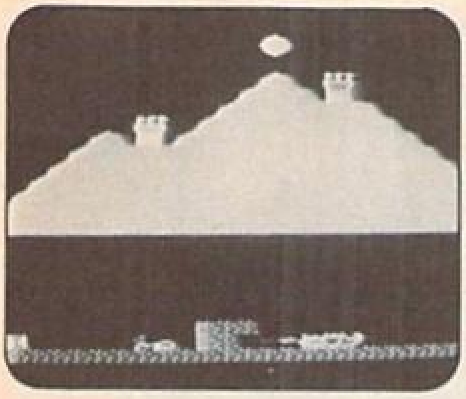
kcul dooG
nosnhoJ eiddE

```
1 REM **** TYPELEFTER ****
2 REM BY EDDIE JOHNSON, 1984
3 REM 2928 TENNESSEE NE,
  ALBUQUERQUE, NM 87110  505)
  299-1662
100 PRINT"[CLEAR]";:X=252:Y=253
110 GET A$:IF A$=""THEN 110
120 P=POS(X):Z=2-P
130 IF A$=CHR$(13)THEN A$=""
    :Y=Y+Z:IF Y<0 THEN Y=0:X=X+Z
140 X=X-1:IF X<0 THEN X=0:Y=Y-1
    :IF Y<0 THEN Y=0:GOTO 100
150 PRINT"[HOME]";SPC(X);
    SPC(Y)A$;
160 GOTO 110
```

Tiger Sprite *(Continued From Page 77)*

```
35 POKE V+0,H:POKE V+1,D
39 POKE V+0,H:POKE V+1,D
50 GET A$:IF A$=""THEN 50
55 IF A$="H"THEN H=H-Z
60 IF A$="J"THEN H=H+Z
70 IF A$="U"THEN D=D-Z
80 IF A$="N"THEN D=D+Z
90 IF A$="Q"THEN PRINT"[CLEAR]"
   :POKE V+21,0:END
95 GOTO 30
100 DATA 2,8,0
110 DATA 1,134,0
120 DATA 10,170,64
130 DATA 10,170,128
140 DATA 43,239,160
150 DATA 25,231,144
160 DATA 42,154,160
170 DATA 25,101,144
180 DATA 10,170,160
190 DATA 2,170,144
200 DATA 10,170,128
210 DATA 38,170,144
220 DATA 25,105,100
230 DATA 42,105,160
240 DATA 21,105,84
250 DATA 42,170,160
260 DATA 26,170,160
270 DATA 10,170,144
280 DATA 1,137,36
290 DATA 2,138,10
300 DATA 10,70,129
310 REM AGE GROUP C:12 TO 18 (14)
```

C



Catapult

For Unexpanded VIC and Datassette

by Joseph Bedard
Hyde Park, Massachusetts

I am 17 years old and attend Boston Technical High School. I started programming about a year ago. I taught myself how to program by looking at other programs—I got most of my knowledge that way.

In my game, "Catapult", you are Prince Valiant—and are fighting for your life against a fire-breathing dragon. Your only weapon against Sindar (the dragon) is a catapult, which has three distance settings. Prince Valiant must run to the rock bin at the left of the screen, bring a rock back to load the catapult and fire it at Sindar.

The prince must hit the dragon squarely on the head before Sindar gets the idea and goes away. No one ever said that dragons are the brightest beasts around—so Valiant must hit Sindar once in the head at level one, twice at level two, three times at level three and so on. Meanwhile, Sindar's fire is destroying the brick wall that is Valiant's only protection, so hurry up with those rocks!

The "Z" key will move the prince leftward to the rock bin, the "C" key takes him back to load the catapult. "F1" launches a *long* shot, "F3" a medium range shot and "F5" a short toss over the wall.

Automatic Load from Disk

For those of you using a VIC 20 with a disk drive, here are the changes you must make to "Catapult Loader" and "Catapult".

Lines 200 to the end of "Catapult Loader" perform some fancy footwork to automatically load the "Catapult" program from the tape. We can make this work for a disk drive by changing line 200 to:

200 LOAD "CATAPULT", 8

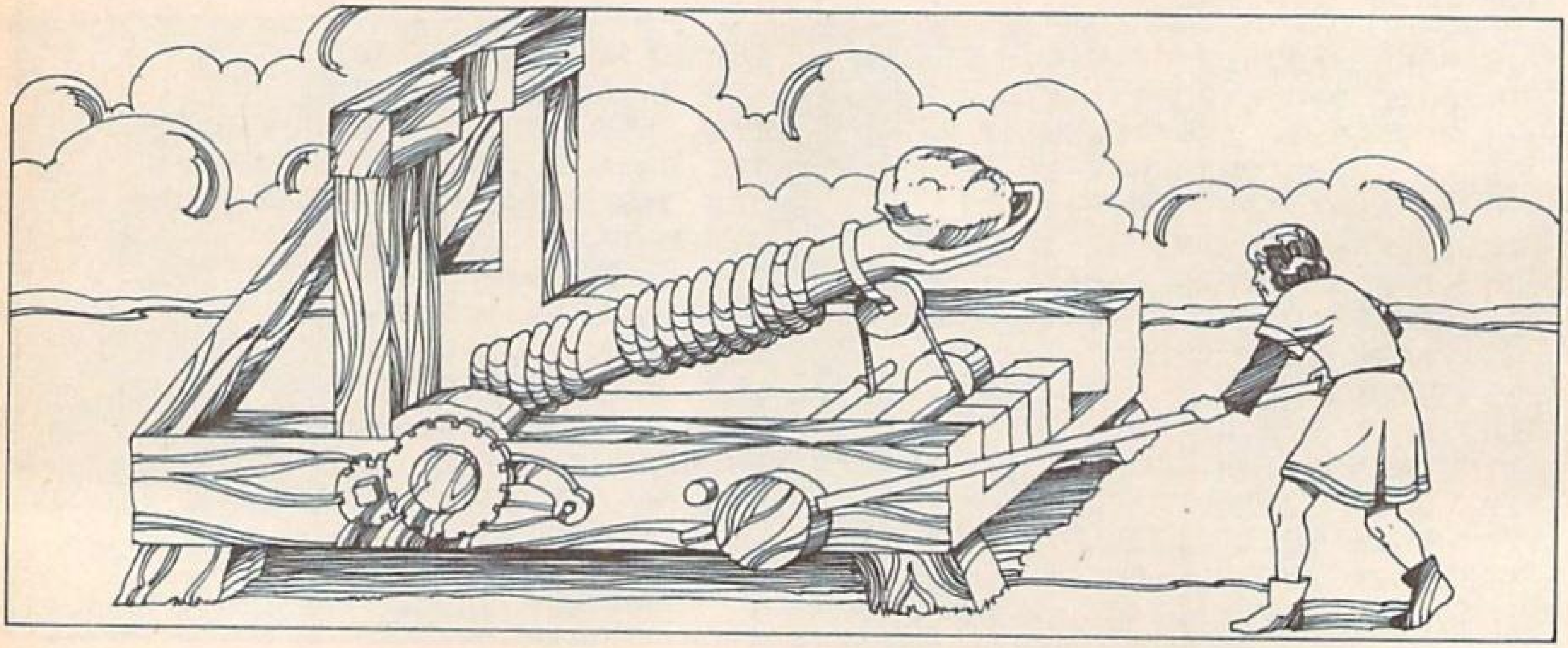
Now we must add a line to the program because normally you can't load a big program from a small program. Add the following line to "Catapult":

1 POKE 45,PEEK(174) : POKE 46,PEEK(175)

When you now run "Catapult Loader", it will automatically load and run "Catapult" from disk.

Catapult Loader

```
5 REM**********************GRAPHI
  CS*FOR*************CATAPULT*****
  **********************
10 PRINT"[CLEAR]":POKE 51,0
   :POKE 52,28:POKE 55,0
   :POKE 56,28:CLR:CB=7168
20 READ A:IF A=-1 THEN 200
30 FOR N=0 TO 7:READ B
```



```
   :POKE CB+A*8+N,B:NEXT:GOTO 20
40 DATA 0,255,129,255,165,165,
   255,129,255
45 DATA 1,24,24,64,56,24,42,68,32
50 DATA 2,24,24,2,28,24,84,34,4
55 DATA 3,0,0,0,0,24,36,36,24
60 DATA 4,0,0,0,28,226,65,255,255
65 DATA 5,96,112,8,4,2,1,255,255
70 DATA 6,3,3,1,1,1,1,255,255
75 DATA 7,72,255,18,255,136,255,
   17,255
80 DATA 8,0,0,92,171,116,0,0,0
85 DATA 9,0,6,13,118,128,241,126,
   7
90 DATA 10,36,109,255,0,193,2,60,
   207
95 DATA 11,134,131,197,57,2,4,
   120,128
96 DATA 12,96,86,45,22,8,17,46,55
100 DATA 32,0,0,0,0,0,0,0,0
110 DATA 13,1,7,15,63,63,127,127,
    255
120 DATA 14,128,240,248,248,252,
    252,254,255
130 DATA 15,255,255,255,255,255,
    255,255,255
135 DATA 16,0,0,1,7,15,31,31,63
140 DATA 17,0,0,128,224,240,248,
    248,252
145 DATA 18,63,31,31,15,7,1,0,0
150 DATA 19,252,248,248,240,224,
    128,0,0
155 DATA 20,16,31,22,29,22,29,22,
    29
160 DATA 21,16,240,208,112,208,
    112,208,112
165 DATA 22,25,29,17,17,42,63,32,
    63
170 DATA 23,216,156,16,16,168,
    248,8,248,-1
200 PRINT"[CLEAR,HOME]"
    :POKE 198,5:POKE 631,78
    :POKE 632,69:POKE 633,87
    :POKE 634,13:POKE 635,131:END
```

Catapult

```
0 REM*****CATAPULT*****
  BY                  JOSEPH BEDARD
2 PRINT"[CLEAR,HOME]"
  :POKE 36879,8:GOSUB 6000
5 QAX=QAX+1
10 PRINT"[CLEAR,HOME]"
   :POKE 36879,8:GOSUB 2000
20 ON M GOTO 40,70
30 M=1:GOTO 20
40 A%=PEEK(197):IF A%=33 THEN
   A=A-1:MA=1:DI=1
45 IF PEEK(A)=0 THEN A=A+1:R=1
   :ZT=3
50 IF A%=34 THEN A=A+1:MA=2:DI=-1
55 IF PEEK(A)=4 AND R=1 THEN R=0
   :M=2:ZT=32
60 IF PEEK(A)=4 THEN A=A-1
65 POKE A,MA:POKE A+C,1
   :POKE A-21,ZT:POKE A+C-21,4
   :POKE A-21+DI,32:POKE A+DI,32
   :GOTO 150
70 IF S=1 THEN 95
71 A%=PEEK(197):IF A%=39 THEN
   F=8022:GOTO 85
75 IF A%=47 THEN F=8043:GOTO 85
80 IF A%=55 THEN F=8064:GOTO 85
81 GOTO 150
85 POKE 36878,15
86 S=1:CA=8149:Q=4:FOR W=1 TO 3
   :POKE CA,Q:POKE 36876,S6:Q=Q+1
   :S6=S6+20
87 POKE CA+C,4:NEXT:POKE 36876,0
90 POKE 36878,5:RO=CA-22:Q=-21
   :S6=201
95 IF RO=F THEN Q=23
   :POKE RO+21,32
100 POKE RO,3:POKE RO+C,4
    :POKE RO-Q,32:POKE A,MA
    :RO=RO+Q
110 IF PEEK(RO)=7 THEN RO=CA-22
    :M=1:S=0:F=0:POKE 8149,4
120 IF PEEK(RO)=9 OR PEEK(RO)=12
    THEN POKE 36878,15
    :POKE 36876,241:Z=Z-1:Z1=Z1+1
    :POKE 36876,0
125 POKE 36878,5
130 IF Z1=QAX THEN 5000
150 ON N GOTO 160,200
155 N=1:GOTO 150
160 IF S1=0 THEN DR=INT(RND(1)*5)
    +8156
170 POKE DR,9:POKE DR+1,10
```

(Continued On Next Page)

```
    :POKE DR+2,11:N=2
200 POKE DR+C,5:POKE DR+1+C,5
    :POKE DR+2+C,5:IF S1=1 THEN
    210
205 POKE DR,12:FB=DR-1:S1=1
    :H1=H1+1
210 POKE 36877,S5:POKE FB,8
    :POKE FB+C,2:POKE FB+1,32
    :FB=FB-1:POKE DR,9:S5=S5-5
215 IF PEEK(FB)=7 AND H1>Z THEN
    POKE FB,32:POKE FB+C,0
    :POKE FB+1,32:FB=DR-1:S1=0
    :H1=0:Y=Y+1
216 IF X>0 THEN 218
217 IF Y=3 THEN X=8129:GOSUB 3000
218 IF Y=6 THEN 5010
220 IF PEEK(FB)=7 THEN POKE FB+1,
    32:FB=DR-1:N=1:S1=0:S5=230
225 IF PEEK(FB)=3 THEN POKE FB+1,
    32:FB=DR-1:N=1:S1=0:S5=230
    :H1=0
230 IF N=1 THEN POKE DR,32
    :POKE DR+1,32:POKE DR+2,32
235 GOTO 20
2000 POKE 36869,255
     :PRINT"[CLEAR]"
2010 PRINT"[UP,SPACE12,YELLOW]PQ
     [SPACE8]"
2020 PRINT"[UP,SPACE12,YELLOW]RS
     [SPACE8]"
2025 PRINT
2030 PRINT"[UP,SPACE12,MAGENTA]
     MN [GREEN]VW[MAGENTA,SPACE5]
     "
2035 PRINT"[UP,SPACE11]MOON
     [GREEN]TU[MAGENTA,SPACE5]"
2040 PRINT"[UP,SPACE10]MOOOOOON
     [SPACE4]"
2045 PRINT"[UP,SPACE6,WHITE]VW
     [MAGENTA] MOOOOOOOON[SPACE3]
     "
2050 PRINT"[UP,SPACE4]MN[WHITE]
     TU[MAGENTA]MOOOOOOOOOON
     [SPACE2]"
2055 PRINT"[UP,SPACE3]
     MOOOOOOOOOOOOOOOON "
2060 PRINT"[UP,SPACE2]
     MOOOOOOOOOOOOOOOOOON"
2065 PRINT"[UP] MOOOOOOOOOOOOOOO
     OOOOOMOOOOOOOOOOOOOOOOOOOOO"
2070 PRINT"[UP]OOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOOOOOOOOOO"
2072 ZT=32:C=30720:FOR A=7988 TO
     8185:POKE A,32:POKE A+C,0
     :NEXT
2085 FOR A=8164 TO 8185:POKE A,7
     :POKE A+C,2:NEXT
2090 POKE 8142,0:POKE 8142+C,7
     :POKE 8149,4:POKE 8149+C,4
2095 FOR A=8129 TO 8131:POKE A,7
     :POKE A+C,2:NEXT
2100 FOR A=8151 TO 8153:POKE A,7
     :POKE A+C,2:NEXT
2110 M=0:N=0:A=8148:MA=1:S5=230
     :POKE 36878,5:S6=201:Z=10
     :Y=0:Z1=0:X=0
2125 RETURN
3000 POKE X,32:POKE X+C,0
     :POKE X+1,32:POKE X+1+C,0
     :POKE X+2,32:POKE X+2+C,0
     :POKE X+22,7
3010 POKE X+22+C,2:POKE X+23,7
     :POKE X+23+C,2:POKE X+24,7
     :POKE X+24+C,2:RETURN
5000 IF Z1=QAX THEN:POKE 36877,0
     :PRINT"[HOME,RVS,DOWN12,
     RIGHT,WHITE]NEXT LEVEL"
     :FOR WTR=1 TO 7500:NEXT
5005 GOTO 5020
5010 POKE 36878,0:POKE 36877,0
     :PRINT"[HOME,RVS,DOWN12,
     RIGHT,GREEN]GAME OVER"
     :FOR WTR=1 TO 7500:NEXT:RUN
5020 S=0:Y=0:X=0:S1=0:FB=0:GOTO 5
6000 POKE 36869,255
     :PRINT"[CLEAR,HOME,DOWN6,
     RIGHT5,RVS,RED] CATAPULT"
6010 C=30720:FOR A=8164 TO 8185
     :POKE A,7:POKE A+C,2:NEXT
6020 PRINT"[DOWN8,RIGHT12,YELLOW]
     PQ[DOWN,LEFT2]RS[RIGHT9,
     DOWN3,WHITE]VW[DOWN,LEFT2]
     TU[HOME]"
6030 A=8159:DR=8161:FOR Z=1 TO 16
6032 POKE A,1:POKE A+C,1
     :POKE DR,9:POKE DR+1,10
     :POKE DR+2,11:POKE DR+C,5
     :POKE DR+1+C,5
6034 POKE DR+2+C,5:POKE A+1,32
     :POKE DR+3,32:A=A-1:DR=DR-1
6040 FOR Y=1 TO 250:NEXT:NEXT
6050 POKE A+1,21:POKE A+1+C,1
     :FOR Y=1 TO 5000:NEXT
6060 Z=0:Y=0:A=0:DR=0:RETURN    C
```

Student Software: Who Owns What a Student Creates?

by Alfred Fant, Jr.

In an academic setting, determining who owns intellectual property can be a rather thorny issue. Should the student own all the proprietary rights to a program he or she creates using school-owned computers?

On the one side of the terminal, we have the student spending weeks of classroom time developing a program for a semester project grade. This programmer makes full use of the provided resources: teacher guidance, debugging tools, computer manuals and the like. And then, if the program is a commercial success, the student sells the software. This "software" consists only of typed words upon sheets of paper. What rights (if any) does the school have to the distribution and profits of such programs?

Some schools claim that since the program could not have been developed without the investment capital in computer resources, the program's profit should be returned to the school. After all, the student programmer has already received a reward—an academic grade. The primary purpose of school is to award grades, not proprietary rights to programs developed during class time on school-owned facilities. Apparently, this is the position of many universities and colleges.

Such assignment of proprietary rights for computer programs is unclear from a legal standpoint. As the May, 1983, issue of *Physics Today* noted:

> ***"The law distinguishes between two types of intellectual property—patents and copyrights. Patents rely on originality and novelty, whereas for copyrights the idea itself doesn't have to be original, only the rendition or expression of the idea. In 1979 the U.S. Supreme Court ruled that software [computer programs] could not be patented; it has not yet ruled on exactly how software should be treated."***

Since many schools are now requiring computer science coursework for graduation, this is a program which needs to be addressed. In the high school where I teach, for example, five student programs were seriously reviewed by various microcomputing magazines. Two programs were finally purchased. The $175.00 received was given totally to the student authors. Perhaps not surprisingly, this was also the first year computer science became a required subject in the student curriculum.

I would like to suggest the following policy toward such student-written programs. Remember, this is only a suggested policy—it is not set in concrete. The private and public school sectors would welcome your suggestions and comments.

Student Computer Program Rights

Whenever a computer program is created by a student using school-owned equipment, the following policies are recommended:

1. The student program should include a personal copyright statement and the name of the school/institution embedded in the actual program code.

example:
copyright 1984 by John A. Doe
Hyde Park Baptist High School
3901 Speedway/Austin TX 78751

This notice would give the student protection from unscrupulous commercialism. Someone might steal a copyright from a student author, but they would probably hesitate stealing a program from a school.

This notice would also give the school valuable public relation benefits. Such PR can be quite helpful in securing monetary grants.

2. The school/institution could have free, unlimited use of the program for educational purpose although the school could not sell or otherwise distribute copies of the program without the author's permission. Thus, the student would actually own the rights to the program.

3. All profits from the program would go to the student. It is suggested that the student voluntarily return 10% of the profit to the school/institution. The school board would decide how to best utilize this 10%. For instance, it could be used to buy more computer equipment or it could be returned to the student. **C**

How to Submit Things to Kids' Corner

Betsy Byrne

Programs: *Must* be sent on disk or tape, clearly marked with computer type and your name, address, and PHONE NUMBER. If your program is for the VIC, mark the label with the amount of memory you had plugged in when you saved the program. Send a printout (listing) if you can, and BE SURE that your disk or tape is mailed in a PADDED protective envelope. Tapes should be in a mailing case, disks between two pieces of stiff cardboard. Protective envelopes are found at office supply stores for around 25 cents. They're worth it!

Articles, Stories, Poems, etc.: Should be typed or computer printed, double spaced, each page numbered, and headed with the title and author's name, address, and phone number. I can, if absolutely necessary, accept text that is neatly hand printed on every other line, each page numbered and headed as described above.

Photographs: Color or black and white, preferably 35mm. Taped or otherwise attached to a stiff piece of paper or cardboard (but make sure they can be removed without damaging them). Label the paper with some type of caption, the name(s) of the person(s) pictured, and the name, address, and phone number of the sender. (School pictures are just fine.)

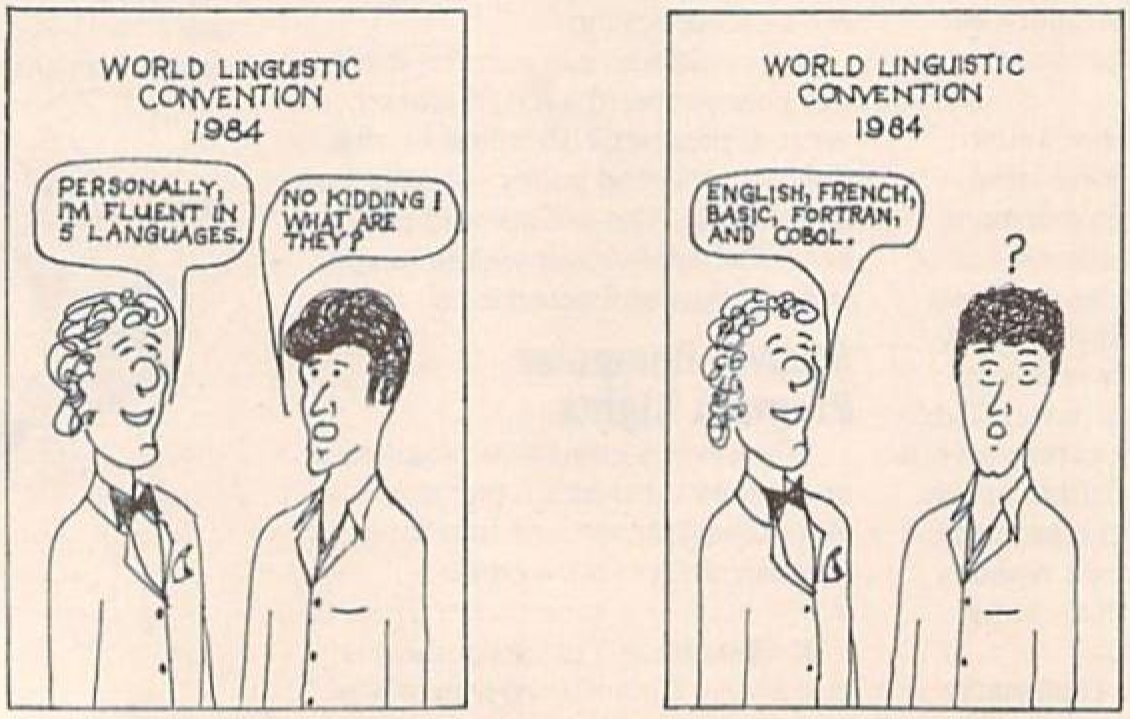
Drawings: Black and white might be best; at least outline things in a dark color if you can. Make them big enough that details will show if they have to be reduced. If you are sending a cartoon, print the captions in the balloons very carefully, or get someone who is really good at printing to do the words for you. Mail them using the same directions as for photographs.

If you are sending a program, don't forget to include the directions! It's a good idea to have them right in the program if you can. Be sure to keep a copy of everything you send. I might need to call you about it, and unfortunately, I can't send anything back. If you have already sent something in and didn't see it in this issue, remember that we have to get *Power/Play* ready for printing almost THREE months before you read it! If something you sent is going to be in the magazine, you will probably get a phone call from me before you see it in print so don't forget to include your AREA CODE and phone number.

Even if you are sending a drawing, photo or program, write something that tells a little bit about YOU. Don't forget to send a picture of yourself!

The address to send things to is:

COMMODORE KIDS
c/o BETSY BYRNE
6209 LESLIE PLACE NE
ALBUQUERQUE NM 87109 **C**

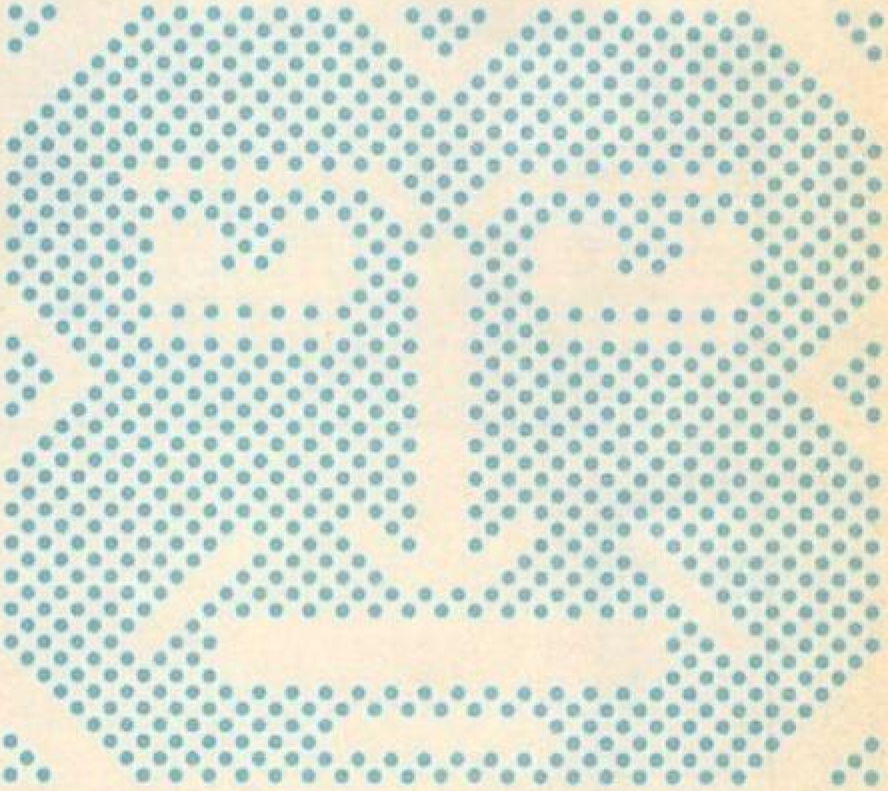

WORLD LINGUISTIC CONVENTION 1984
PERSONALLY, I'M FLUENT IN 5 LANGUAGES.
NO KIDDING! WHAT ARE THEY?
WORLD LINGUISTIC CONVENTION 1984
ENGLISH, FRENCH, BASIC, FORTRAN, AND COBOL.
?

Greg Mavko
Edmonton, Alberta

Borderline and Type Setter

by Richard K. Stare

The 1525 graphic printer can be useful for many things besides listing programs and printing letters. The following two programs are an interesting way to expand that usefulness.



Borderline

"Borderline" is a poor man's way to distinctive stationery using the VIC 1525 graphic printer. Written to run on a VIC with at least 3K expansion, "Borderline" is a simple, malleable program that dot-programs the printer to generate a border around a page of tractor paper. For the purpose of example, the listing given is already set up to produce a photographic motif—little cameras, shown front and side view, are printed around the edge of the paper using this version. However, you can modify it to produce any design you like.

Despite its length, the program is essentially nothing more than a few simple repetitive operations. Twenty-six characters are built up and stored in string variables during the first 41 lines of the program. The rest of the program fits the variables into a pattern a line at a time and then prints the line. The pattern, placement of character variables, character data requirements and data line numbers relating to the variables are shown in Figures 1 and 2.

Modifying the program for different character designs is a simple matter. The first step is the art work. Using a fine-grid graph paper, mark off the character variable borders in the manner of Figure 1. Now create a design within the confines of the layout. Keep in mind that the character sizes are fixed; altering the sizes would necessitate a restructuring of the entire program.

Inserting the design into the program requires some thought and a little effort. Referring to Figure 1, determine the binary value for each column of the character variables. For example, in the camera design illustrated in Figure 1, column one of character variable BA$ is 255, column two is 255, column three is 131 and so on for 12 columns, the length of the character. See the 1525 printer manual or the *VIC 20 Programmer's Reference Guide* for full details on binary conversion.

Figure 2 gives the program line number of the data statement defining each character variable for entering a new design. For example, once the binary values for BA$ are known, you can refer to Figure 2 to see that program line number seven should be replaced with the new data statement reflecting this data. When the data has all been entered test run the program and check for errors.

To lengthen or shorten the border just add or subtract steps from line 58. To decrease the width subtract one or more steps from the FOR-TO loops that construct the master strings P1$, P2$, P3$, and P4$ and decrease the dot address given in variable SP$ by 23 for each step deleted.

Type Setter

In order to facilitate data development for the "Borderline" program, I wrote another program, titled "Type Setter". Since the 26 characters used requires 309 binary numbers to define them it is helpful to have a program that does the figuring.

"Type Setter" is an aid to generating character data for the VIC 1525 graphic printer in graphic mode. The idea is similar to Commodore's custom character tape but with a few twists. Designed for the VIC, it requires a minimum of 3K to operate.

Input to the program is via screen and joystick, but with modification a light pen can be substituted for the joystick. The screen work area consists of three segments, each representing 18x7 dots. Dots are set or unset by positioning the cursor with the joystick and

Figure 1.
Borderline Character Designing

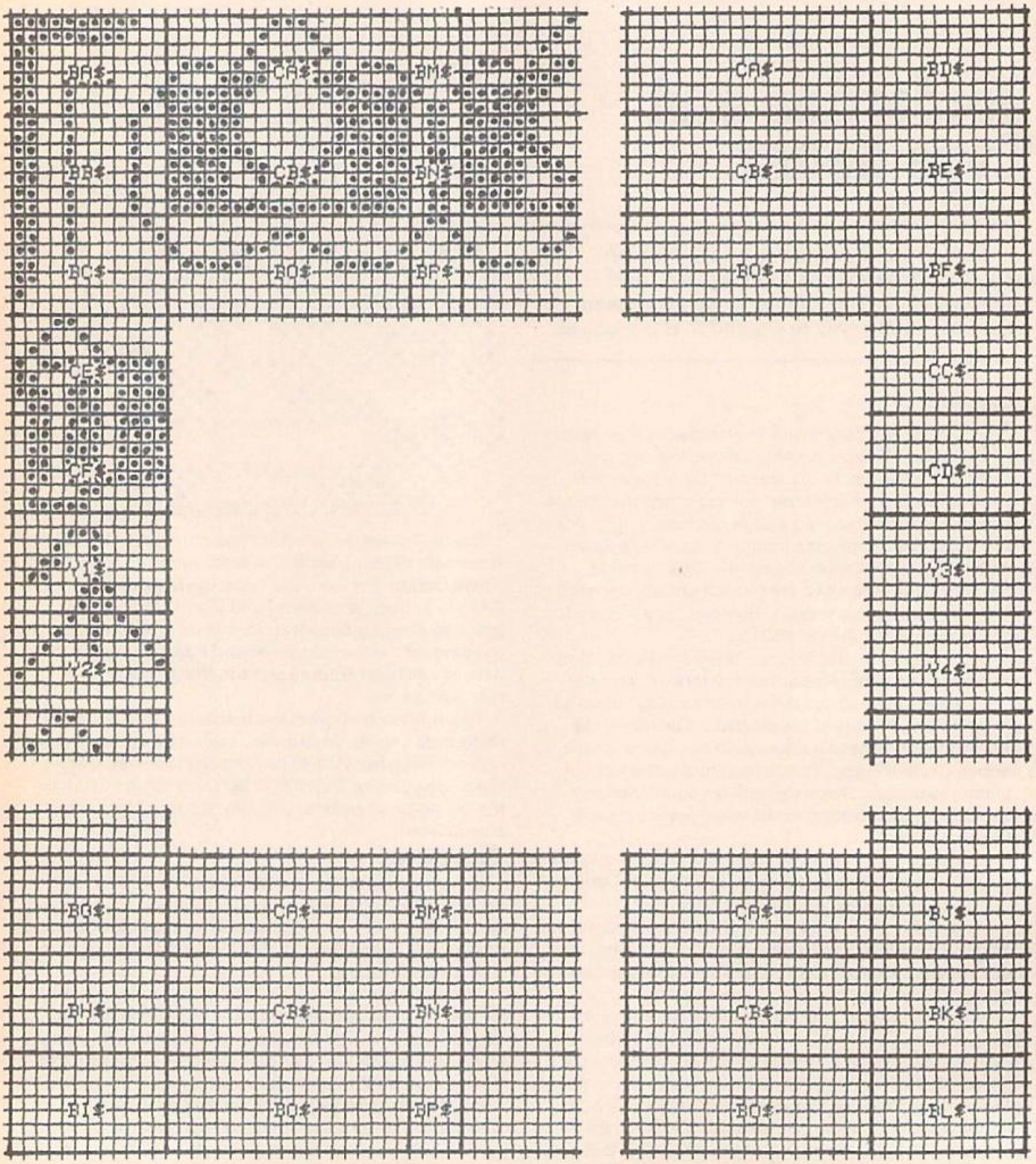

BA$
CA$
BM$
BB$
CB$
BN$
BC$
BO$
BF$
CE$
CF$
Y1$
Y2$
CA$
BD$
CB$
BE$
BO$
BF$
CC$
CD$
Y3$
Y4$
BG$
CA$
BM$
BH$
CB$
BN$
BI$
BO$
BP$
CA$
BJ$
CB$
BK$
BO$
BL$

Figure 2. *Borderline Character Data*

Page Line No.	Character Placing
1-4	Space
5	BA$+ 19*(CA$+ BM$)+ CA$+ BD$
6	BB$+ 19*(CB$+ BN$)+ CB$+ BE$
7	BC$+ 19*(BO$+ BP$)+ BO$+ BF$
8	CE$+ SP$+ CC$
9	CF$+ SP$+ CD$
10	Y1$+ SP$+ Y3$
11	Y2$+ SP$+ Y4$
12-91	Lines 8-11 repeated
92	BG$+ 19*(CA$+ BM$)+ CA$+ BJ$
93	BH$+ 19*(CB$+ BN$)+ CB$+ BK$
94	BI$+ 19*(BO$+ BP$)+ BO$+ BL$
95-99	Space

Note: 99 lines to the page (8½ × 11 inch paper).

Character Size and Program Line No.

Char.	X 7 Dots	Data Stat.
BA$	12 dots	Line 7
CA$	19 dots	Line 1
BM$	4 dots	Line 19
BD$	12 dots	Line 10
BB$	12 dots	Line 8
CB$	19 dots	Line 2
BN$	4 dots	Line 20
BE$	12 dots	Line 11
BC$	12 dots	Line 9
BO$	19 dots	Line 22
BP$	4 dots	Line 21
BF$	12 dots	Line 12
CE$	12 dots	Line 5
CC$	12 dots	Line 3
CF$	12 dots	Line 6
CD$	12 dots	Line 4
Y1$	12 dots	Line 23
Y3$	12 dots	Line 25
Y2$	12 dots	Line 24
Y4$	12 dots	Line 26
BG$	12 dots	Line 13
BJ$	12 dots	Line 16
BH$	12 dots	Line 14
BK$	12 dots	Line 17
BI$	12 dots	Line 15
BL$	12 dots	Line 18

Note: SP$ is dot address for righthand border.

pressing the fire button. Function commands are turned on or off when the cursor is over the first letter of the command at screen bottom and the fire button is pressed. Moving the cursor to one of the numbers to the left or right of screen and pressing the fire button will activate whichever command is on at the time. Completion of a command cycle turns that command off.

Program output is contained within the program itself. Prior to a run, blank data statements using the format illustrated in lines 101 to 103 are entered into the program. Upon activation of the save command, the designated segment is scanned, converted to printer binary, the first available data blank located and the binary data POKEd into the blank. To use the data, delete the program body as well as the last two bytes (,0) of each statement. The data statements can then be manipulated and merged with the intended program. Obviously, the Programmer's Aid cartridge will simplify the procedure.

"Type Setter" notes:

Lines 1 to 3 compute the location of the first blank data statement for any memory configuration. If the program is modified so that it is shorter, the value added to K to define U must be changed. Lengthening the program will only increase the time it takes to complete the first save.

Line 11 prevents the program from reading the fire button too soon after an entry.

Lines 25 to 33 are the delete function.

Lines 35 to 43 convert data scanned from the screen to printer binary.

Lines 45 to 48 locate the first available blank data statement with 47 ending the program should all statements be full.

Lines 49 to 51 POKE the printer data into the located data statement.

Lines 101 to 103 should be duplicated as needed. The extra comma is a result of the way the data is POKEd in. The extra zero is to remind that the comma is there.

Borderline

```
1 DATA 240,232,232,232,240,248,
  180,154,137,137,137,154,180,
  248,240,232,232,232,240
2 DATA 255,255,255,255,255,199,
  204,216,208,208,208,216,204,
  199,255,255,255,255,255
3 DATA 248,248,248,248,136,252,
  242,233,137,234,244,184
4 DATA 159,159,159,159,144,191,
  255,223,192,223,255,128
5 DATA 184,244,234,137,233,242,
  252,136,248,248,248,248
6 DATA 128,255,223,192,223,255,
  191,144,159,159,159,159
7 DATA 255,255,131,131,243,147,
  147,147,131,129,192,160
8 DATA 255,255,128,128,255,128,
  128,128,128,255,128,128
9 DATA 191,159,128,128,135,128,
  128,128,128,128,129,130
10 DATA 160,192,129,131,147,147,
   147,243,131,131,255,255
11 DATA 128,128,255,128,128,128,
   128,255,128,128,255,255
```

(Continued On Next Page)

```
12 DATA 130,129,128,128,128,128,
   128,135,128,128,159,191
13 DATA 254,252,128,128,240,128,
   128,128,128,128,192,160
14 DATA 255,255,128,128,255,128,
   128,128,128,255,128,128
15 DATA 255,255,224,224,231,228,
   228,228,128,128,129,130
16 DATA 160,192,128,128,128,128,
   128,240,128,128,252,254
17 DATA 128,128,255,128,128,128,
   128,255,128,128,255,255
18 DATA 130,129,128,128,228,228,
   228,231,224,224,255,255
19 DATA 160,192,192,160
20 DATA 128,255,255,128
21 DATA 130,129,129,130
22 DATA 132,136,136,136,136,136,
   132,132,130,130,130,132,132,
   136,136,136,136,136,132
23 DATA 128,144,152,140,130,226,
   254,224,128,128,128,128
24 DATA 144,136,132,130,129,128,
   143,128,129,130,132,136
25 DATA 128,128,128,128,224,254,
   226,130,140,152,144,128
26 DATA 136,132,130,129,128,143,
   128,129,130,132,136,144
27 SP$=CHR$(27)+CHR$(16)+CHR$(1)
   +CHR$(212)
28 FOR I=1 TO 19:READ A
   :CA$=CA$+CHR$(A):NEXT
   :FOR I=1 TO 19:READ A
   :CB$=CB$+CHR$(A):NEXT
29 FOR I=1 TO 12:READ A
   :CC$=CC$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :CD$=CD$+CHR$(A):NEXT
30 FOR I=1 TO 12:READ A
   :CE$=CE$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :CF$=CF$+CHR$(A):NEXT
31 FOR I=1 TO 12:READ A
   :BA$=BA$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :BB$=BB$+CHR$(A):NEXT
32 FOR I=1 TO 12:READ A
   :BC$=BC$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :BD$=BD$+CHR$(A):NEXT
33 FOR I=1 TO 12:READ A
   :BE$=BE$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :BF$=BF$+CHR$(A):NEXT
34 FOR I=1 TO 12:READ A
   :BG$=BG$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :BH$=BH$+CHR$(A):NEXT
35 FOR I=1 TO 12:READ A
   :BI$=BI$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :BJ$=BJ$+CHR$(A):NEXT
36 FOR I=1 TO 12:READ A
   :BK$=BK$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :BL$=BL$+CHR$(A):NEXT
37 FOR I=1 TO 4:READ A
   :BM$=BM$+CHR$(A):NEXT
   :FOR I=1 TO 4:READ A
   :BN$=BN$+CHR$(A):NEXT
38 FOR I=1 TO 4:READ A
   :BO$=BO$+CHR$(A):NEXT
39 FOR I=1 TO 19:READ A
   :BP$=BP$+CHR$(A):NEXT
40 FOR I=1 TO 12:READ A
   :Y1$=Y1$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :Y2$=Y2$+CHR$(A):NEXT
41 FOR I=1 TO 12:READ A
   :Y3$=Y3$+CHR$(A):NEXT
   :FOR I=1 TO 12:READ A
   :Y4$=Y4$+CHR$(A):NEXT
42 OPEN 3,4:FOR U=1 TO 4
   :PRINT#3,CHR$(8):NEXT
43 FOR I=1 TO 5:P1$=P1$+CA$+BM$
   :NEXT:FOR I=1 TO 5
   :P2$=P2$+CA$+BM$:NEXT
   :FOR I=1 TO 5:P3$=P3$+CA$+BM$
   :NEXT
44 FOR I=1 TO 4:P4$=P4$+CA$+BM$
   :NEXT
45 PRINT#3,CHR$(8)BA$P1$P2$;
46 PRINT#3,P3$P4$CA$BD$
47 P1$="":P2$="":P3$="":P4$=""
48 FOR I=1 TO 5:P1$=P1$+CB$+BN$
   :NEXT:FOR I=1 TO 5
   :P2$=P2$+CB$+BN$:NEXT
   :FOR I=1 TO 5:P3$=P3$+CB$+BN$
   :NEXT
49 FOR I=1 TO 4:P4$=P4$+CB$+BN$
   :NEXT
50 PRINT#3,CHR$(8)BB$P1$P2$;
51 PRINT#3,P3$P4$CB$BE$
52 P1$="":P2$="":P3$="":P4$=""
53 FOR I=1 TO 5:P1$=P1$+BP$+BO$
   :NEXT:FOR I=1 TO 5
   :P2$=P2$+BP$+BO$:NEXT
   :FOR I=1 TO 5:P3$=P3$+BP$+BO$
   :NEXT
54 FOR I=1 TO 4:P4$=P4$+BP$+BO$
```

```
   :NEXT
55 PRINT#3,CHR$(8)BC$P1$P2$;
56 PRINT#3,P3$P4$BP$BF$
57 P1$="":P2$="":P3$="":P4$=""
58 FOR U=1 TO 21
59 PRINT#3,CE$SP$CC$
60 PRINT#3,CF$SP$CD$
61 PRINT#3,Y1$SP$Y3$
62 PRINT#3,Y2$SP$Y4$
63 NEXT
64 P1$="":P2$="":P3$="":P4$=""
65 FOR I=1 TO 5:P1$=P1$+CA$+BM$
   :NEXT:FOR I=1 TO 5
   :P2$=P2$+CA$+BM$:NEXT
   :FOR I=1 TO 5:P3$=P3$+CA$+BM$
   :NEXT
66 FOR I=1 TO 4:P4$=P4$+CA$+BM$
   :NEXT
67 PRINT#3,CHR$(8)BG$P1$P2$;
68 PRINT#3,P3$P4$CA$BJ$
69 P1$="":P2$="":P3$="":P4$=""
70 FOR I=1 TO 5:P1$=P1$+CB$+BN$
   :NEXT:FOR I=1 TO 5
   :P2$=P2$+CB$+BN$:NEXT
   :FOR I=1 TO 5:P3$=P3$+CB$+BN$
   :NEXT
71 FOR I=1 TO 4:P4$=P4$+CB$+BN$
   :NEXT
72 PRINT#3,CHR$(8)BH$P1$P2$;
73 PRINT#3,P3$P4$CB$BK$
74 P1$="":P2$="":P3$="":P4$=""
75 FOR I=1 TO 5:P1$=P1$+BP$+BO$
   :NEXT:FOR I=1 TO 5
   :P2$=P2$+BP$+BO$:NEXT
   :FOR I=1 TO 5:P3$=P3$+BP$+BO$
   :NEXT
76 FOR I=1 TO 4:P4$=P4$+BP$+BO$
   :NEXT
77 PRINT#3,CHR$(8)BI$P1$P2$;
78 PRINT#3,P3$P4$BP$BL$
79 FOR U=1 TO 4:PRINT#3:NEXT
   :CLOSE 3
```

Type Setter

```
0 DD=37154:PA=37137:PB=37152
  :DIM BI(17):DA=101
1 W=INT(PEEK(43))/16:G=INT(W)/16
  :Y=INT(PEEK(44))/16:Z=INT(Y)/16
2 L=W-INT(W):C=L*16:O=G-INT(G)
  :D=O*16:P=Y-INT(Y):E=P*16
  :M=Z-INT(Z):F=M*16
3 G=C*1:H=D*16:I=E*(16*16)
  :J=F*(16*16*16):K=G+H+I+J
  :U=K+2242
4 SC=4096:CC=37888:IF
  PEEK(36869)AND 32 THEN SC=7680
  :CC=38400
5 POKE 36879,8:PRINT"[CLEAR]";
  :FOR I=1 TO 21:PRINT"[SPACE2,
  RVS,WHITE,SHFT @18,RVOFF]":NEXT
6 POKE SC+66,49:POKE CC+66,1
  :POKE SC+87,49:POKE CC+87,1
  :POKE SC+155,99:POKE CC+155,1
  :POKE SC+174,99
7 POKE CC+174,1:POKE SC+220,50
  :POKE CC+220,1:POKE SC+241,50
  :POKE CC+241,1:POKE SC+309,99
  :POKE CC+309,1
8 POKE SC+328,99:POKE CC+328,1
  :POKE SC+374,51:POKE CC+374,1
  :POKE SC+395,51:POKE CC+395,51
9 POKE SC+463,99:POKE CC+463,1
  :POKE SC+482,99:POKE CC+482,1
10 PRINT"[DOWN,SPACE2,RVS,WHITE]
   S[RVOFF]AVE[SPACE2,RVS]D
   [RVOFF]ELETE[SPACE2,RVS]Q
   [RVOFF]UIT[HOME]"
11 FOR I=1 TO 100:NEXT
12 GOSUB 52:PE=PEEK(LO)
   :IF PE>128 THEN POKE LO,PE-128
   :GOTO 14
13 POKE LO,PE+128
14 IF FR AND PE=250 THEN PE=170
   :POKE LO,PE:GOTO 11
15 IF FR AND PE=170 THEN PE=250
   :POKE LO,PE:GOTO 11
16 IF FR AND PE=147 THEN PE=19
   :Q=1:POKE LO,PE:GOTO 11
17 IF FR AND PE=19 THEN PE=147
   :Q=0:POKE LO,PE:GOTO 11
18 IF FR AND PE=132 THEN PE=4
   :QQ=1:POKE LO,PE:GOTO 11
19 IF FR AND PE=4 THEN PE=132
   :QQ=0:POKE LO,PE:GOTO 11
20 IF FR AND PE>48 AND PE<52
   THEN GOSUB 23
21 IF FR AND PE=145 THEN POKE
   36879,27:PRINT"[CLEAR,BLUE]"
```

(Continued On Next Page)

```
    :END
22 POKE LO,PE:GOTO 12
23 IF Q=0 AND QQ=0 THEN RETURN
24 IF Q>0 THEN 34
25 IF PE=49 THEN 28
26 IF PE=50 THEN 30
27 IF PE=51 THEN 32
28 PRINT"[HOME]";:FOR I=1 TO 7
   :PRINT"[RIGHT2,RVS,WHITE,
   SHFT @18,RVOFF]":NEXT
   :POKE SC+492,132
29 QQ=0:RETURN
30 PRINT"[HOME,DOWN6]"
   :FOR I=1 TO 7:PRINT"[RIGHT2,
   RVS,WHITE,SHFT @18,RVOFF]"
   :NEXT:POKE SC+492,132
31 QQ=0:RETURN
32 PRINT"[HOME,DOWN13]"
   :FOR I=1 TO 7:PRINT"[RIGHT2,
   RVS,WHITE,SHFT @18,RVOFF]"
   :NEXT:POKE SC+492,132
33 QQ=0:RETURN
34 V=1:B(0)=V:FOR I=1 TO 6:V=V+V
   :B(I)=V:NEXT:POKE SC+486,147
   :FOR I=0 TO 17:BI(I)=0:NEXT
35 IF PE=49 THEN 38
36 IF PE=50 THEN 40
37 IF PE=51 THEN 42
38 FOR Y=0 TO 6:I=0:FOR X=2 TO 19
   :IF PEEK(SC+X+22*Y)=170 THEN
   BI(I)=BI(I)+B(Y)
39 I=I+1:NEXT X:NEXT Y
   :FOR I=0 TO 17:BI(I)=BI(I)+128
   :NEXT:GOTO 44
40 FOR Y=7 TO 13:I=0
   :FOR X=2 TO 19:IF
   PEEK(SC+X+22*Y)=170 THEN
   BI(I)=BI(I)+B(Y-7)
41 I=I+1:NEXT X:NEXT Y
   :FOR I=0 TO 17:BI(I)=BI(I)+128
   :NEXT:GOTO 44
42 FOR Y=14 TO 20:I=0
   :FOR X=2 TO 19:IF
   PEEK(SC+X+22*Y)=170 THEN
   BI(I)=BI(I)+B(Y-14)
43 I=I+1:NEXT X:NEXT Y
   :FOR I=0 TO 17:BI(I)=BI(I)+128
   :NEXT:GOTO 44
44 Q=0
45 IF PEEK(U-3)=DA AND
   PEEK(U-2)=0 AND PEEK(U-1)=131
   AND PEEK(U)=48 THEN 49
46 IF PEEK(U-3)>DA AND
   PEEK(U-2)=0 AND PEEK(U-1)=131
   AND PEEK(U)=48 THEN
   DA=PEEK(U-3):GOTO 49
47 IF PEEK(U-3)=0 AND
   PEEK(U-2)=0 AND PEEK(U-1)=0
   THEN PRINT"[RVS]DATA FULL
   [RVOFF]":END
48 U=U+1:GOTO 45
49 FOR I=0 TO 17:N$=STR$(BI(I))
50 FOR II=2 TO 4:NN$=MID$(N$,II,
   1):POKE U,ASC(NN$):U=U+1
   :NEXT II
51 POKE U,44:U=U+1:NEXT I:DA=DA+1
   :RETURN
52 EP=PEEK(37139):POKE 37139,0
53 POKE DD,127:S3=-((PEEK(PB)AND
   128)=0):POKE DD,255:P=PEEK(PA)
   :S1=-((P AND 8)=0)
   :S2=((P AND 16)=0)
54 S0=((P AND 4)=0)
   :FR=-((P AND 32)=0):X=S2+S3
   :Y=S0+S1:POKE 37139,EP
55 X1=X1+X:IF X1<0 OR X1>21 THEN
   X=X*-1:GOTO 55
56 Y1=Y1+Y:IF Y1<0 OR Y1>22 THEN
   Y=Y*-1:GOTO 56
57 LO=SC+X1+22*Y1:RETURN
101 DATA 000,000,000,000,000,000,
    000,000,000,000,000,000,000,
    000,000,000,000,000,0
102 DATA 000,000,000,000,000,000,
    000,000,000,000,000,000,000,
    000,000,000,000,000,0
103 DATA 000,000,000,000,000,000,
    000,000,000,000,000,000,000,
    000,000,000,000,000,0
```

C

Print Gothic or Bold

on Your VIC 1525 Printer

by Brent Miller

A few months ago I saw something written in gothic and I wondered if it was possible to program my VIC 1525 printer to print out a gothic set. After some thinking and playing around I discovered a very easy way to do this. This program allows anyone to type in a letter or whatever and, by pressing the desired function keys, have the printer print it out in either gothic, bold or standard as single- or double-width characters.

Before I go any further, I should briefly describe how the VIC 1525 prints out user-defined characters. The printer can print a maximum height of only seven dots but the width is unlimited. The easiest way to get the data for the characters is to start with a grid that is seven dots high by eight or nine wide. Then number along the right side of the grid from the top to the bottom as follows:

```
    +---+---+---+---+---+---+---+---+
 1  |   | X | X | X |   |   |   |   |
    +---+---+---+---+---+---+---+---+
 2  | X |   |   |   | X |   |   |   |
    +---+---+---+---+---+---+---+---+
 4  | X |   |   |   | X |   |   |   |
    +---+---+---+---+---+---+---+---+
 8  | X | X | X | X | X |   |   |   |
    +---+---+---+---+---+---+---+---+
 16 | X |   |   |   | X |   |   |   |
    +---+---+---+---+---+---+---+---+
 32 | X |   |   |   | X |   |   |   |
    +---+---+---+---+---+---+---+---+
 64 | X |   |   |   | X |   |   |   |
    +---+---+---+---+---+---+---+---+
      A   B   C   D   E   F   G   H
```

You can see that this is a graph of the standard letter "A". The next step is to add up each vertical row. In this case, row A is 2+4+8+16+32+64=126. Now add 128 to this number and you get 254. This is the first data number for the letter "A". Continue the process for the next eight columns (I chose eight columns instead of six in order to create the bold set). The final data statement for the letter "A" is: DATA 254, 137, 137, 137, 254, 0, 0, 0.

Now the way in which the data must be sent to the printer is: read the data one at a time; take the chr$ of the number; add all of these up; open a channel to the printer; print chr$(8) to put the printer in the graphic mode and print the variable containing the information. Here is an example:

```
10 FOR I=1 TO 8
20 READ X
30 X$=X$+CHR$(X): REM IF DOUBLE WIDTH ADD
CHR$(X) AGAIN
40 NEXT I
50 OPEN 1,4 : PRINT#1, X$
99 DATA 254, 137, 137, 137, 254, 0, 0, 0
```

Although this will print out only the letter "A", if you redesign the graph to be another character, then that character will be printed out.

Now that you understand how to print out user-defined characters, we will go to the two problems I encountered in writing this program. The first problem was how to take in the characters that the user prints on the screen and not to accept some of them, like the CLEAR/HOME key.

The easiest way I found to do this is to use the GET statement and if the character was a legal one (one that needs to be printed) then print it to the screen and add one to the array of characters. This way, everything that is typed is stored in an array, say x$(x). If the word "it" was typed, then x$(1)="i" and x$(2)="t".

Line 212 shows how to print all of the characters in memory on the screen with the use of a loop. When the computer is in the execution mode the cursor is off. To turn the cursor on you must POKE zeros into locations 204 and 207, which will keep the cursor blinking most, but not all, of the time (line 140).

Now that we have the characters stored in an array how do we determine which set of data to print out for each character? My first thought was 26 IF/THEN statements that would test the character in the array to see if it equaled "A", "B", "C", etc... Then print out the corresponding data for that character. Well, the only problem with this method is that it would probably take an hour to print out a short letter! So I had to think of some other way to do it. I finally came up with a method that works almost as fast as printing out standard characters!

To do the program I noticed that the ASC code of letters A-Z was 65-90. The whole key to the program is to initially read the data into an array starting at 65 and ending with 90. Let's say that we read the data into the array gt$(i) as "i" goes from 65 to 90. Now all the data for the letter "A" is in gt$(65) and all the data for "Z" is in gt$(90).

Let's say the characters that the user typed in went into the array x$(m), where "m" is the number of characters that were typed in. If the word "it" was typed, x$(1)="i" and x$(2)="t". If we take the ASC value of x$(1) we would get the value for "I" which is 73. All we need to do now is use the 73 as the array index of gt$(i) and send this to the printer, which will print gt$(73)—the data corresponding to "i". The actual print statement would look like this:

```
10 FOR X=1 TO M
20 PRINT# 1,CHR$(8)GT$(ASC(X$(X)))
30 NEXT M
```

This will print the corresponding data for all the characters that were typed. By first checking the ASC codes of the characters we can determine if the special data for that character should be sent to the printer or if the character itself should be sent. For example, numbers are not changed in the gothic set that I designed and therefore can be sent to the printer directly. In the program, this is done on lines 480-540.

If you are wondering why the data statements have fifteen numbers, this is because I put all the data for both the gothic set and the bold on the same line. On lines 20-90 are the READ statements and the variables used are:

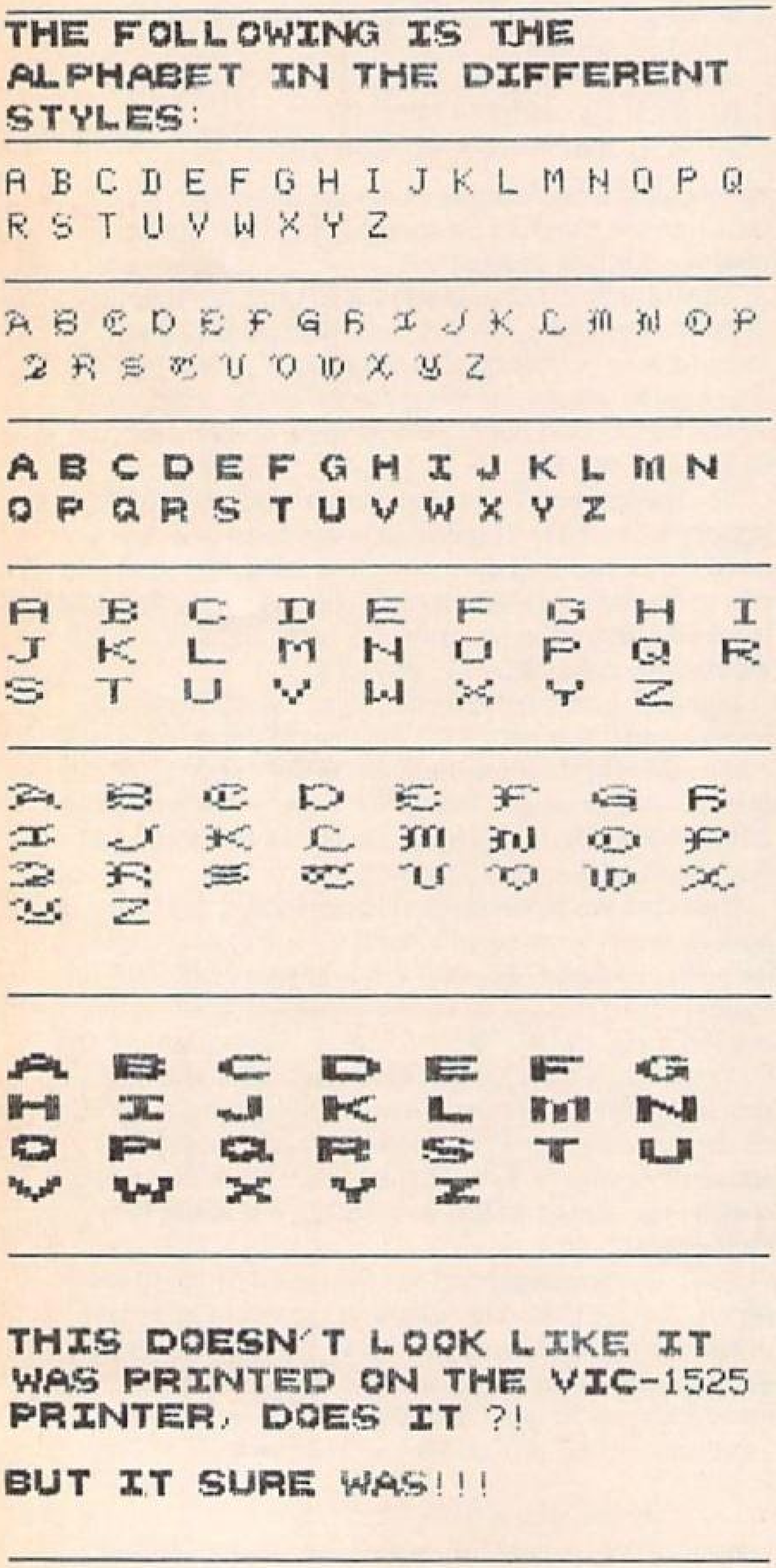
gt$(i)==> Gothic single width
dg$(i)==> Gothic double width
bo$(i)==> Bold single width
db$(i)==> Bold double width

The first seven numbers are for the gothic set, while the other eight numbers are for the bold set.

In using the program, if you are unsure at any time about what is on the screen press the F2 key. This will show you what is in memory at this time. To delete a character use the DELETE key because the cursor keys are turned off. The line across the top of the screen has a "B" and a "G" on it. This is used to determine one line of print on the printer in the single width mode. For example, the "B" is about halfway on the screen, which means that one and a half lines of print equal one line of bold print on the printer. The same holds true for the gothic set.

The nice thing about this program is that anyone can now write a custom set of data and replace this data with their data. The next project is to combine this program with a powerful word processor to make it even easier to use! Figure 1 is a sample of the different type styles. Have fun!! **C**

Figure 1.
Custom Characters on the 1525 Printer



THE FOLLOWING IS THE ALPHABET IN THE DIFFERENT STYLES:

A B C D E F G H I J K L M N O P Q
R S T U V W X Y Z

A B C D E F G H I J K L M N O P
Q R S T U V W X Y Z

A B C D E F G H I J K L M N
O P Q R S T U V W X Y Z

A B C D E F G H I
J K L M N O P Q R
S T U V W X Y Z

A B C D E F G H
I J K L M N O P
Q R S T U V W X
Y Z

A B C D E F G
H I J K L M N
O P Q R S T U
V W X Y Z

THIS DOESN'T LOOK LIKE IT WAS PRINTED ON THE VIC-1525 PRINTER, DOES IT ?!

BUT IT SURE WAS!!!

Gothic/Bold

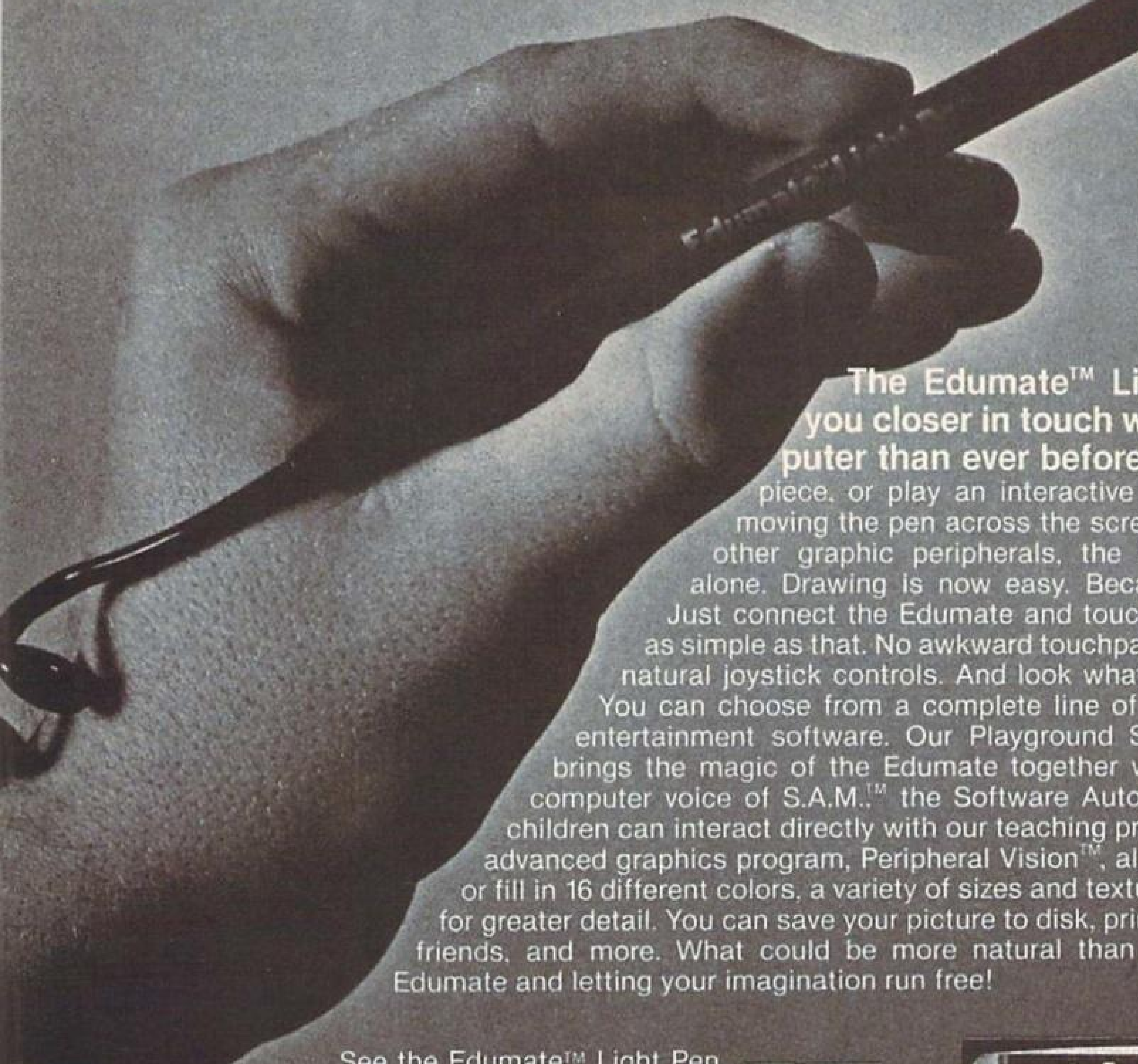
```
1 REM PROGRAM PRINT FONTS BY
  BRENT MILLER
5 PRINT"[CLEAR] PLEASE WAIT..."
10 DIM X$(1000), DG$(90),
    DB$(90), GT$(90), BO$(90)
20 FOR I=65 TO 90
30 FOR J=1 TO 7
40 READ GT:GT$(I)=GT$(I)+CHR$(GT)
   :DG$(I)=DG$(I)+CHR$(GT)+CHR$
   (GT)
50 NEXT J
60 FOR J=1 TO 8
70 READ BO:BO$(I)=BO$(I)+CHR$(BO)
   :DB$(I)=DB$(I)+CHR$(BO)+CHR$
   (BO)
80 NEXT J
90 NEXT I
95 PRINT"[CLEAR]F2 - PRINTS OUT
   DOCUMENT ON[SPACE18]
   THE SCREEN."
96 PRINT
97 PRINT"F7 - PRINTS DOCUMENT ON
```

```
    THE[SPACE18]PRINTER,
     BUT DOES NOT"
 98 PRINT"[SPACE5]CLEAR IT FROM
    MEMORY.":PRINT
 99 PRINT"F8 - CLEARS THE
    DOCUMENT FROM[SPACE16]MEMORY."
100 PRINT:PRINT"[RVS]
    PRESS ANY CHARACTER TO BEGIN
    [RVOFF]"
110 M=1
120 GET I$:IF I$=""THEN 120
130 PRINT"[CLEAR,RVS]BEGIN
    :[RVOFF]":HD=0
135 FOR F=1 TO 40:PRINT"[RVS] ";
    :NEXT F:PRINT"[UP]
    "TAB(20)"B"TAB(30)"G"
140 PRINT:PRINT:POKE 204,0
    :POKE 207,0:POKE 53280,0
    :POKE 53281,0:B=1:D=1
150 GET X$(M):IF X$(M)=""OR
    X$(M)="[UP]"OR X$(M)="[DOWN]
    "OR X$(M)="[RIGHT]
    "OR X$(M)="[LEFT]"THEN 150
160 IF ASC(X$(M))>31 AND
    ASC(X$(M))<128 THEN PRINT
    X$(M);:M=M+1:GOTO 150
170 IF ASC(X$(M))>159 AND
    ASC(X$(M))<192 THEN PRINT
    X$(M);:M=M+1:GOTO 150
180 IF ASC(X$(M))=13 THEN PRINT
    X$(M);:M=M+1:GOTO 150
185 HD=0
190 IF PEEK(197)=0 THEN PRINT"
    [LEFT] [LEFT]";:HD=1:M=M-1
    :IF M<1 THEN M=1:GOTO 130
195 IF HD=1 THEN 150
200 IF X$(M)="[F7]"THEN 240
210 PRINT"[CLEAR]":FOR F=1 TO 40
    :PRINT"[RVS] ";:NEXT F
    :PRINT"[UP]"TAB(20)"B"TAB(30)
    "G":
212 FOR X=1 TO M:PRINT X$(X);
    :NEXT X
220 IF X$(M)="[F8]"THEN M=1
    :GOTO 130
230 GOTO 150
240 PRINT"[RVS,CLEAR]
    WHAT PRINT SETTINGS DO YOU
    WANT?[SPACE5]":PRINT:PRINT
250 PRINT"[RVS,SPACE3]
    F1 - SELECT CHARACTER SET
    [SPACE10]":PRINT
260 PRINT"[RVS,SPACE3]
    F3 - SELECT WIDTH[SPACE18]"
    :PRINT
265 PRINT"[RVS,SPACE3]
    F4 - RETURN TO EDIT[SPACE16]"
    :PRINT
270 PRINT"[RVS,SPACE3]
    F5 - PRINT OUT[SPACE21]"
    :PRINT:PRINT
280 GET ZZ$:IF ZZ$=""THEN 280
290 IF ZZ$="[F1]"THEN B=B+1
    :IF B>3 THEN B=1
295 IF ZZ$="[F1]"THEN 330
300 IF ZZ$="[F3]"THEN D=D+1
    :IF D>2 THEN D=1
305 IF ZZ$="[F3]"THEN 360
310 IF ZZ$="[F5]"THEN 380
315 IF ZZ$="[F4]"THEN 210
320 GOTO 280
330 IF B=1 THEN PRINT"[RVS,UP,
    SPACE2]STANDARD SET":GOTO 280
340 IF B=2 THEN PRINT"[RVS,UP,
    SPACE2]BOLD SET[SPACE4]"
    :GOTO 280
350 IF B=3 THEN PRINT"[RVS,UP,
    SPACE2]GOTHIC SET[SPACE2]"
    :GOTO 280
360 IF D=1 THEN PRINT"[RVS,UP,
    SPACE2]SINGLE WIDTH":GOTO 280
370.IF D=2 THEN PRINT"[RVS,UP,
    SPACE2]DOUBLE WIDTH":GOTO 280
380 REM ***PRINT ROUTINE***
390 OPEN 1,4
400 IF(B=1)AND(D=1)THEN 460
410 IF(B=1)AND(D=2)THEN 470
420 IF(B=2)AND(D=1)THEN 480
430 IF(B=2)AND(D=2)THEN 510
440 IF(B=3)AND(D=1)THEN 540
450 IF(B=3)AND(D=2)THEN 570
460 FOR X=1 TO M:PRINT#1,
    CHR$(15)X$(X);:NEXT X
    :PRINT#1,CHR$(13):CLOSE 1
    :GOTO 210
470 FOR X=1 TO M:PRINT#1,
    CHR$(14)X$(X);:NEXT X
    :PRINT#1,CHR$(13):CLOSE 1
    :GOTO 210
480 FOR X=1 TO M:IF
    ASC(X$(X))>64 AND
    ASC(X$(X))<91 THEN 500
490 PRINT#1,CHR$(15)X$(X);:NEXT X
    :PRINT#1,CHR$(13):CLOSE 1
    :GOTO 210
500 PRINT#1,CHR$(8)BO$(ASC(X$(X))
    );:NEXT X:PRINT#1,CHR$(13)
    :CLOSE 1:GOTO 210
510 FOR X=1 TO M:IF
    ASC(X$(X))>64 AND
```

(Continued On Next Page)

```
    ASC(X$(X))<91 THEN 530
520 PRINT#1,CHR$(14)X$(X);:NEXT X
    :PRINT#1,CHR$(13):CLOSE 1
    :GOTO 210
530 PRINT#1,CHR$(8)DB$(ASC(X$(X))
    );:NEXT X:PRINT#1,CHR$(13)
    :CLOSE 1:GOTO 210
540 FOR X=1 TO M:IF
    ASC(X$(X))>64 AND
    ASC(X$(X))<91 THEN 560
550 PRINT#1,CHR$(15)X$(X);:NEXT X
    :PRINT#1,CHR$(13):CLOSE 1
    :GOTO 210
560 PRINT#1,CHR$(8)GT$(ASC(X$(X))
    );:NEXT X:PRINT#1,CHR$(13)
    :CLOSE 1:GOTO 210
570 FOR X=1 TO M:IF
    ASC(X$(X))>64 AND
    ASC(X$(X))<91 THEN 590
580 PRINT#1,CHR$(14)X$(X);:NEXT X
    :PRINT#1,CHR$(13):CLOSE 1
    :GOTO 210
590 PRINT#1,CHR$(8)DG$(ASC(X$(X))
    );:NEXT X:PRINT#1,CHR$(13)
    :CLOSE 1:GOTO 210
9000 DATA 194,185,149,146,148,
     184,192,224,252,158,155,155,
     158,252,224
9002 DATA 221,170,201,201,201,
     182,128,255,255,235,235,235,
     255,182,128
9004 DATA 156,162,221,193,197,
     162,128,156,190,227,227,227,
     227,162,128
9006 DATA 255,162,193,193,162,
     156,128,255,255,227,227,227,
     227,190,156
9008 DATA 221,170,201,193,193,
     164,128,255,255,235,235,235,
     235,227,128
9010 DATA 201,190,137,137,129,
     130,128,255,255,155,155,155,
     131,131,128
9012 DATA 156,178,169,169,169,
     250,128,156,190,227,193,235,
     187,250,128
9014 DATA 254,137,137,201,178,
     128,128,255,255,140,140,140,
     255,255,128
9016 DATA 196,162,162,190,162,
     145,128,128,227,227,255,255,
     227,227,128
9018 DATA 160,208,192,192,160,
     158,129,176,240,224,224,255,
     191,128,128
9020 DATA 201,190,136,148,162,
     193,128,255,255,140,156,182,
     227,193,128
9022 DATA 192,190,193,193,194,
     192,160,255,255,224,224,224,
     224,128,128
9024 DATA 201,190,129,190,129,
     190,128,255,255,130,252,252,
     130,255,255
9026 DATA 201,190,130,188,192,
     191,128,255,255,134,140,152,
     176,255,255
9028 DATA 156,162,221,193,193,
     162,156,128,190,247,227,227,
     247,190,128
9030 DATA 201,190,137,137,137,
     134,128,255,255,155,155,155,
     142,142,128
9032 DATA 194,197,227,241,210,
     172,128,128,190,247,227,243,
     183,254,224
9034 DATA 201,190,137,137,145,
     230,128,255,255,155,155,155,
     255,238,128
9036 DATA 196,170,170,170,170,
     145,128,174,207,219,217,217,
     251,178,128
9038 DATA 154,166,154,198,194,
     198,161,128,131,131,255,255,
     131,131,128
9040 DATA 130,129,190,192,192,
     190,129,191,255,224,224,224,
     255,191,128
9042 DATA 130,129,158,162,193,
     161,158,135,158,184,224,184,
     158,135,128
9044 DATA 129,190,192,190,194,
     162,156,143,255,240,156,156,
     240,255,143
9046 DATA 194,161,146,156,166,
     193,160,128,227,247,156,156,
     247,227,128
9048 DATA 130,161,206,208,208,
     206,177,128,135,143,248,248,
     143,135,128
9050 DATA 128,225,209,201,197,
     195,128,128,227,243,255,239,
     231,227,128                   C
```

how to really get in touch with your computer.


Edumate Light Pen

The Edumate™ Light Pen puts you closer in touch with your computer than ever before. Paint a masterpiece, or play an interactive game simply by moving the pen across the screen. Compared to other graphic peripherals, the Edumate stands alone. Drawing is now easy. Because it's natural. Just connect the Edumate and touch the screen. It's as simple as that. No awkward touchpad surface. No unnatural joystick controls. And look what else is at hand. You can choose from a complete line of educational and entertainment software. Our Playground Software™ series brings the magic of the Edumate together with the amazing computer voice of S.A.M.,™ the Software Automatic Mouth, so children can interact directly with our teaching programs. Our new advanced graphics program, Peripheral Vision™, allows you to draw or fill in 16 different colors, a variety of sizes and textures, and zoom in for greater detail. You can save your picture to disk, print a copy for your friends, and more. What could be more natural than picking up an Edumate and letting your imagination run free!

See the Edumate™ Light Pen at your local computer store or call direct 1-800-334-SOFT. Available for Commodore 64™ & Atari® computers. Please specify computer, memory, and disk/cassette when ordering.

Edumate Light Pen
64

Drawn with Edumate™ Light Pen

Little Red Riding Hood
playground software

Alphabet Construction Set
playground software

Animal Crackers
playground software

Computer Crayons
playground software

futurehouse™

P.O. Box 3470 Department C.
Chapel Hill, North Carolina 27514

Edumate Light Pen, Playground Software, Computer Crayons, Alphabet Construction Set are trademarks of Futurehouse, Inc.
S.A.M. is a trademark of Don't Ask Software.
ks of Commodore Electronics, Ltd. and Atari, Inc., respectively.

Tape Tips

by Alfred Fant, Jr.

The bell is ringing. You have a five minute passing period between classes, but your next class is COMPUTERS!—and you are there in less than a minute. Every second counts and you are so anxious to get started programming. Your assignment (a personal adventure game based on nested subroutines) is not due for another two weeks, but... well... programming is so much fun. You can't believe it's educational!

You pop your cassette tape into the tape drive, turn on the TV monitor, the Commodore 64 microcomputer, and then hastily type in the magical passwords of computer programming: LOAD "MYPROGRAM". You hit the PLAY button on the tape drive and you wait. Five seconds later, the screen says FOUND MYPROGRAM and you hit the space bar (or C= key) to begin the actual loading of the program into the computer.

And then it happens...

The cassette tape drive stops by itself and the demonical message of computer programming appears:

```
?LOAD ERROR
READY
```

"Arrggghh! Mr. Fant, come here quick... pleeze!" you cry. Your computer science teacher looks tired. Apparently, he has seen this same problem before and he offers these words of wisdom: "Rewind the tape, type in NEW, type in LOAD (by itself, without a specific program name), and try loading it again." You do, it doesn't, and the same demonical message returns to the screen to taunt you.

Sound familiar? As the teacher in the above scenario, I see such problems many times every day and they always seem to affect the same students. Another student using the same machine will have no problems whatsoever. Thus, it seems logically that most ?LOAD ERRORs are student dependent. In other words, it is usually the student's misuse of equipment that causes the ?LOAD ERRORs to appear.

We use both tape and diskette in our classroom as a matter of choice. As one programming assignment is completed, the students who were on tape driven computer systems will switch to diskette driven computer systems. Those formerly using diskette program storage will switch to cassette tape program storage. This policy allows all the students to experience both sequential data tape storage and random diskette data storage. It is after our switchover that an amazing thing becomes clear. *The same students who had ?LOAD ERRORs with the tapes tend to have I/O errors with the diskettes.*

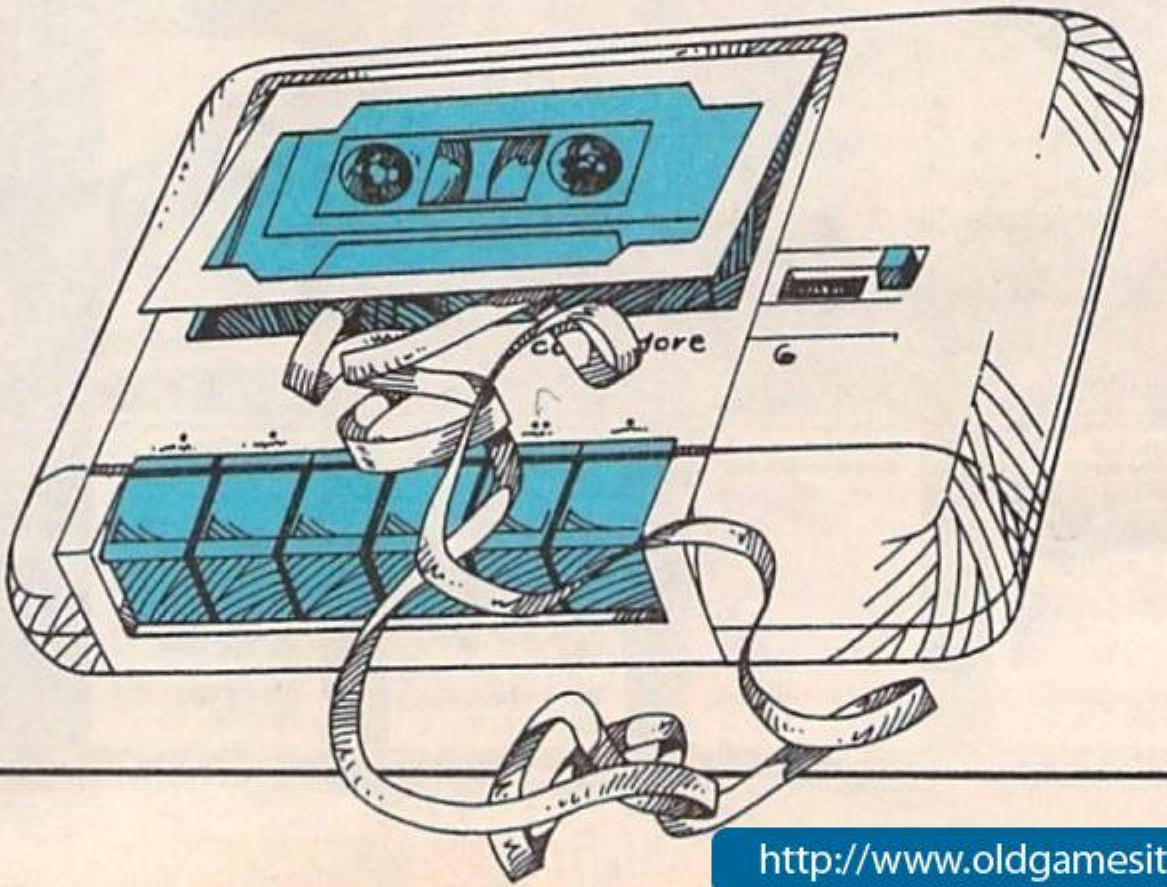
"Arrggghh! Mr. Fant, come quick... my tape's being eaten by this stupid tape recorder!" Jennifer King at station seven is frantically waving her hands, as if she's drowning in a sea of snarled tapes.

"Arrggghh! Mr. Fant, come quick... my program won't stop running. None of the keys have any effect anymore!" Another plaintive cry from John Johanson at station twelve.

These calls for help may seem unique to each victim, but their solutions are really quite common. However, much of my classroom-inspired "keyboard magic" is not documented in any of the Commodore publications. The reason? These solutions solve many machine error problems, but are not foulproof. You can lose your program. Even so, these partial solutions can seem miraculous to those victims of their own sloppy programming habits.

An Ounce of Prevention...

"Never-never-never," I repeat to the students, "should you turn on your TV monitor with your cassette tape within three feet of it." Electromagnetic radiations from the TV monitor can erase bit patterns from your cassette tape. The loss of even a little bit can render your tape program unreadable. This is why there is such a long cord between the cas-



sette recorder and your microcomputer and why it is shielded. The Commodore cassette tape drive manual recommends a minimum separation of 24 inches between it and the TV monitor. Place the cassette tape drive on the floor and keep it there. The cord is long enough to allow this precaution.

"Be sure to press the REC/PLAY buttons (not the REC/REWIND buttons) on the cassette tape drive when you are saving a program." If you do accidentally press REC/REWIND, tap the RUN/STOP key, rewind the tape, advance the tape past the leader and try again to save your program.

"Always VERIFY your programs stored on tape before you shut the Commodore 64 microcomputer off at the end of class." If your program does not verify, rewind the tape and save it again. See your Commodore manual for instructions on using the VERIFY command.

"Never leave your cassette tape drive with the PLAY button depressed when you are programming." Ever leave a cassette tape in a car's stereo system overnight? When you do, the tape is under constant pressure from the pinch roller, and the rubber pinch roller itself (over time) becomes deformed. This deformed pinch roller can cause any tape drive to eat your tape for breakfast.

"Never PLAY or REC/PLAY a cassette tape before assuring *all* slack in the tape has been removed by the takeup reel." A simple twist of the finger is all it takes. If you don't do this, the odds are very high the excess tape will end up inside the pinch roller mechanism instead of where it belongs on the reel.

"Always clean/demagnetize the cassette tape drives at least once every ten hours of usage." This rule is self-explanatory and cleaning kits are available at many retail electronic stores.

"Never use the INS/DEL key to erase to the *left* of a BASIC program line number." If you do, the odds are very great you will trap your machine in an infinite loop. Use the two cursor keys at the bottom right of the keyboard to move around on the screen.

"Always make backup copies of programs that are important to you." Even when a program is being created, save a backup copy for every screenful of program code.

Specific Cures

Program in infinite loop: Press the RUN/STOP key down firmly. LIST your program.

Program in infinite loop— RUN/STOP key is frozen: Holding the RUN/STOP key down firmly, tap the RESTORE key. LIST your program.

Program in infinite loop— RUN/STOP, RESTORE keys are frozen: Remove your cassette tape from the tape drive. Holding the SHIFT key down firmly, tap the # key. Press PLAY on the tape drive, wait three seconds, and then tap the RUN/STOP key. LIST your program.

Program in infinite loop—all keys are frozen: If you have a Commodore printer connected to the Commodore 64, cycle the power of the printer. Wait at least five seconds between turning the printer on/off. LIST your program.

None of the above: Turn the Commodore 64's power off and rely upon your backup tape/disk copy. You *did* save a backup copy, didn't you?

?LOAD ERROR—the first, second and third times: Type NEW, rewind the tape and try again. If no errors, LIST your program.

?LOAD ERROR—the fourth time: Thoroughly clean the tape drive and try again. Type NEW, rewind the tape and type in LOAD (by itself, without a specific program name). Observe (using the tape counter) the exact position of the ?LOAD ERROR. If no errors, LIST your program.

?LOAD ERROR—the fifth time: Try again, but press the RUN/STOP key just prior to the position of the ?LOAD ERROR. If no error, LIST your program.

?LOAD ERROR—the sixth time: Give it up and rely on your backup copy.

These are some of the tricks used in one classroom to help recover student programs bitten by machine errors. A still more powerful tool is my "Tape Doctor" program, which recovers the good data from a tape with a ?LOAD ERROR. Watch for a future article in this magazine.

NEVER UNDERSOLD
—Leader Since 1980—
COMPUSETTE
at
Manufacturer Prices
Industries Finest

	Retail	Your Cost
C-05	99¢	55¢
C-10	$1.29	59¢
C-20	$1.59	69¢
C-30	$1.89	89¢
Cases	49¢	21¢

Packed 24 Per Pack
UPS $3.00 Per Pack 48 States
Micro-80™ INC.
2665 E. Busby Road
Oak Harbor, Wash. 98277
VISA MasterCard
IMMEDIATE SHIPPING
(206) 675-6143

COMMODORE
-USER WRITTEN SOFTWARE-
Supporting all COMMODORE computers
Written by users, for users
★ GAMES ★ UTILITIES ★ EDUCATIONAL ★

VIC 20™
Vic 20 collectins #1, 2, 3, 4, 5, 6
over 70 programs per collection - Tape/Disk - $10.00
Vic 20 collections #7, 8
over 50 programs per collection - Tape/Disk - $10.00

COMMODORE 64™
64 collection #1, 2, 3, 4, 5, 6, 7,
over 25 programs per collection - Tape/Disk - $10.00

COMMODORE 64 COLLECTION #8
GAMES—Action Hockey ● Backgammon ● Civil War ● Dixie 64 ● Have a Nice Day ● Mission to Mars ● Octopus 1 ● Race Saucer Invasion ● UTILITIES—64 Label ● Alpha Index Dates ● Decision Maker ● Disassembler ● Event Calendar Find Nom. Int. ● Income Tax Adv ● Investment F.V. Music Driver IN ● Music Driver P1 ● Music Driver P2 Music Driver P3 ● Music Driver P4 ● Pareto ● Roman Numerals ● Seq. Append ● Term 64 ● Voice 3 Music EDUCATIONAL—A Fable ● Curve Fit ● Finance Calc. ● Foil Inst ● Foil Pract ● Heat Solver ● Intergration ● Percent Drill Projectile ● Titrator 64 ● Weigh

PET® / CBM®
22 collections - Tape/Disk - $10.00 each

DINSET™: Reset Switch
Works on Vic 20 or Commodore 64 – $5.00

SERIAL CABLES
10Ft.—$10.00 15Ft.—$15.00

LOC-LITE™
Operation Status Indicator Assembled & Tested
$20.00

All prices include shipping and handling.
CHECK, MONEY ORDERS,
VISA and MASTERCARD accepted.

For A Free Catalog Write:
Public Domain, Inc.
5025 S. Rangeline Rd., W. Milton, OH 45383
10:00 a.m. - 5:00 p.m. EST – Mon. thru Fri.
(513) 698-5638 or (513) 339-1725

[illegible]

Editing Characters on the 64

by Keith Golden

A very nice thing about the Commodore character set is that it can be changed. A new alphabet of italic or boldface print can be created, and special symbols, such as a degrees sign or an accent can be made. And, of course, the more traditional game graphics, like flying spaceships and bug-eyed aliens can be devised. Graphics can be made for shoot-em-ups, or fancy lettering for adventure games, even foreign alphabets; the limit is your imagination. This is possible because of a location in memory known as the Memory Control Register of the VIC chip, the lower nybble of which controls where VIC gets its dot-matrix for the characters it puts on the screen. By changing this location, we can re-program all of the 510 characters available in Commodore's upper and lower case character sets. This, of course is not exclusive to Commodore, nor is it privileged information that only the *elite* programmers know. It is, however, a great source of frustration for many beginners who find it both difficult to calculate the data involved and mundane to type it in. For this frustration, the *Programmer's Reference Guide,* as well as many other authoritative books, is largely to blame. While calculating data and typing the numbers into DATA statements is a good technique for creating a very small number of characters, doing that for any more than ten becomes tedious and time-consuming. Of course, the way a true programmer deals with something that is tedious and time-consuming is to write a program that does it for him.

Naturally, there are many good programs on the market that help in editing characters. You might want to buy one of these, or even create your own. But if you would rather not shell out the bucks for a professional character-editing program, and you don't feel quite up to the hours at the keyboard needed to produce your own, then I suggest typing in the program at the end of this article.

As this program stands, it requires a joystick and a disk drive to use. If you don't have one of these, turn to the section titled *Editing the Editor*. The other two sections, *Using the Editor*, and *How it Works*, should give you some idea of what this program can do, and how it is accomplished.

Using the Editor

Upon running the program, you will be prompted with the question:

SWITCH IN CHARACTERS? (Y/N)

An affirmative answer will cause the computer to copy all of its ROM characters into a special RAM location set aside for that purpose. Regretfully, this is a slow process, and you will have to wait about a minute for it to be completed. I suggest saving the standard character set on a separate disk (one that you don't plan to have your own special characters SAVEd on) by hitting 'S' when the machine restores control back to you. This way you'll only have to "switch" the characters in once, because after that, you can LOAD them from your disk, which is far quicker.

When the computer has finished copying its character set, it sets the pointer to the new characters and clears the screen. It then displays all 255 characters of the upper-case character set (Figure 1). (Simply pressing the SHIFT and COMMODORE keys at the same time will display the lower-case characters and cause the computer to deal with these.) A solid, non-blinking "cursor" will be positioned in the upper left corner of the screen. This cursor is what you use to select your character. Simply position the cursor on the character of your choice, and hit the fire button. The screen will clear, and a grid (Figure 2) will appear. Moving the X cursor onto the grid and hitting the fire button will reverse the status of that particular bit (i.e. 1=0 and 0=1). This is simpler than using two keys to turn bits on and off, and it's easier to keep track of. The actual character is on the left of the grid, so you can watch the changes as you work on it.

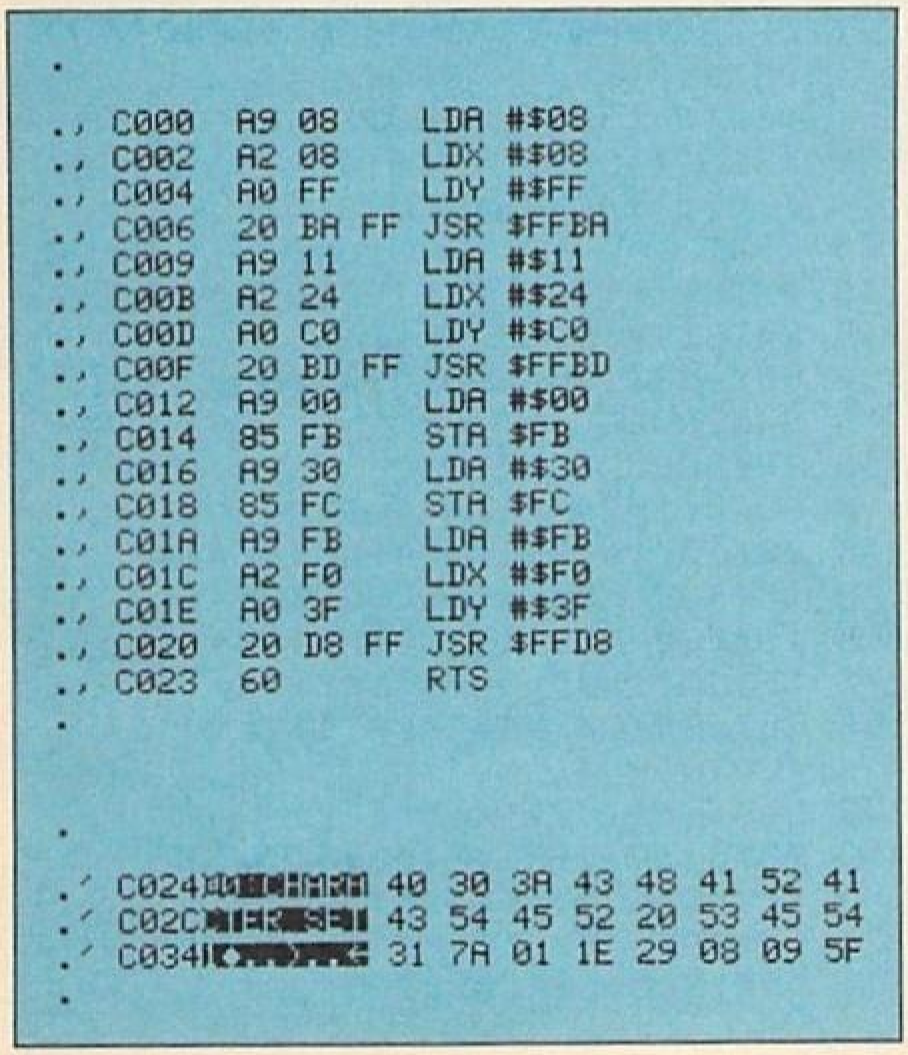
After your character is complete, hit the F7 key to return to the menu. At this point you have the option of saving your place on disk , or LOADing a previous character set in. This is done by hitting 'S' or 'L' respectively. The program SAVEs using the "SAVE with Replace" feature, so you can update the character set as often as you like. But don't hit 'L' unless you have a set on disk to LOAD in, as a drive error will result that this program is unable to handle.

How It Works

There are a few features of this program that may not be very obvious to the layman. The first is the way the program is stored on disk. Line 400 POKEs a machine language program into memory from the DATA lines 310 to 340. The program resides in location 49152 ($C000), the beginning of RAM. The subroutine is in ML rather than BASIC in order to use the Kernal SAVE routine, which SAVEs your characters the way a machine language program would be SAVEd, and allows you to LOAD it back the same way. The main reason for all this is that it SAVEs the file using fewer blocks on disk, taking less time to LOAD and SAVE, and making the whole process much neater and more convenient. I tend to dislike integrating BASIC and ML when it is not necessary, but this is a case where it is.

In order to re-LOAD the character set, just type 'LOAD"character set1",8,1'. This tells the computer to LOAD character set1 at the address pointed to by the headers on the disk ($C000). 'POKE 53272,(PEEK(53272) AND240)+12' will tell the Memory Control Register to get its information from your character set. Those two commands are all you need to type in. If the characters were stored in a file, you would need a complicated program just to put them in memory.

You may wonder why I didn't use the Kernal LOAD routine to LOAD the characters back into memory. The main reason is that there is no need to do so. But also, there is a subtlety in LOADing from a BASIC program that you had best be aware of if you plan to LOAD anything from a program. The best way to demonstrate this is to use BASIC in *my* program and show you how *I* did it. When BASIC encounters the command to LOAD in a program, it realizes (or rather the designers realized) that if it LOADs right over the program it's LOADing from, any command to RUN the program afterward won't be executed, and the computer will return to command mode, leaving the user somewhat bewildered. In order to make it possible for one program to call another, BASIC sends in a command to RUN right after a LOAD from program mode. This is taken a step further, and variables are left intact in order to allow parameters to be passed from one program to the next. This makes it very simple for us. After we LOAD the character set from disk, BASIC wants to RUN it, but all it does is restart our program. By putting a flag somewhere at the beginning, we can direct BASIC safely back to where it left off after LOADing. This flag is usually in line zero or one, but you will find it in line 390, which is where I branched off to. (As you can see, I am no advocate of structured programming. Functional programming is about all I can *handle*.) In line eight, the variable 'Q' is set to one and is tested every time the program restarts. If the result is true, it branches to line 12; if not, it continues.



```
.
.,  C000  A9 08     LDA #$08
.,  C002  A2 08     LDX #$08
.,  C004  A0 FF     LDY #$FF
.,  C006  20 BA FF  JSR $FFBA
.,  C009  A9 11     LDA #$11
.,  C00B  A2 24     LDX #$24
.,  C00D  A0 C0     LDY #$C0
.,  C00F  20 BD FF  JSR $FFBD
.,  C012  A9 00     LDA #$00
.,  C014  85 FB     STA $FB
.,  C016  A9 30     LDA #$30
.,  C018  85 FC     STA $FC
.,  C01A  A9 FB     LDA #$FB
.,  C01C  A2 F0     LDX #$F0
.,  C01E  A0 3F     LDY #$3F
.,  C020  20 D8 FF  JSR $FFD8
.,  C023  60        RTS
.


.
.'  C024          40 30 3A 43 48 41 52 41
.'  C02C          43 54 45 52 20 53 45 54
.'  C034          31 7A 01 1E 29 08 09 5F
.
```

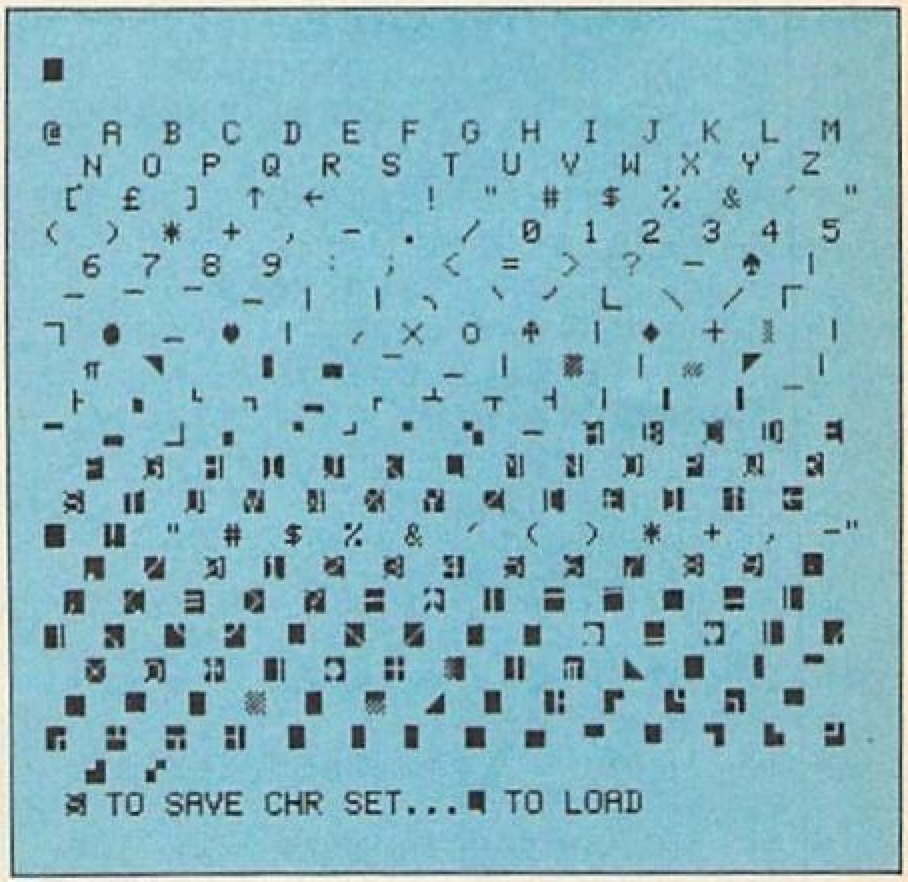
Editing the Editor

Here are some tips on how to make changes in the editor.

It was written for disk, but will work with tape by changing the fourth number of DATA in line 310, which should be an eight, to a one. Also change line 43 to:

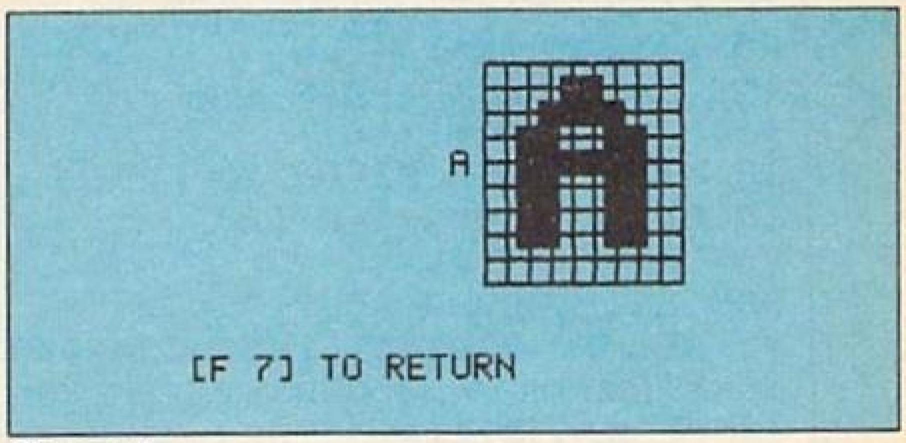
43 IF g$="1" THEN LOAD "character set1",1,1

If you don't have a joystick, the lines to change are 25 to 28 and 170 to 195, substituting the '(joy AND x)=0' for 'g$="(cursor direction)"'. I won't show all the changes, because they are all pretty much the same. Also make similar changes for lines 41 and 220, which detect the



TO SAVE CHR SET... TO LOAD

Figure 1



[F 7] TO RETURN

Figure 2

fire button. I suggest you use the space bar for the fire button and cursors for the various directions, but, of course, you may use whatever keys you want.

The program doesn't input a file name from the user when it SAVEs. This is for convenience. If you want to have more than one file on the same disk, just increment the last number, a 49, in DATA line 340, to 50; and change the file name in line 43 to "character set2". (50 is the ASCII code for "2")

Of course, you may make any changes you like to this program. I think you will find the program simple, after examining it, but infinitely helpful, and I encourage you to make any changes that would make it more so. If you feel that all those characters are more than you need, dig in there and trim it down to your size. I hope I have given some good pointers here for anyone who would like to alter this program, or make one of his own.

Character Editor

```
0 REM *** CHARACTER  EDITOR ***
1 REM *** BY  KEITH  GOLDEN ***
2 GOSUB 390
3 POKE 53281,1:POKE 53280,1
  :REM WHITE BACKGROUND
4 PRINT"[CLEAR,DOWN3,RIGHT,BLACK]
  SWITCH IN CHARACTERS? (Y/N)"
5 GET AN$:IF AN$="Y"THEN 900
6 IF AN$="N"THEN 8
7 GOTO 5
8 POKE 53272,(PEEK(53272)AND
  240)+12:REM TURN ON CHRSET
9 B$="[RIGHT14]":A$=B$+"[BLACK,
  DOWN7]":Q=1
10 PRINT"[CLEAR]":FOR T=0 TO 255
   :POKE 1104+T*3,T
   :POKE 55376+T*3,0:NEXT
   :REM DISPLAY CHRS
11 PRINT"[HOME,DOWN] [RVS]S
   [RVOFF] TO SAVE CHR SET...
   [RVS]L[RVOFF] TO LOAD"
12 P=32:PO=1024
15 REM BEGIN CURSOR ROUTINE
20 JOY=PEEK(56320)
21 POKE PO,P1
25 IF(JOY AND 1)=0 THEN Y=Y-1
   :POKE PO,P:J=1
26 IF(JOY AND 2)=0 THEN Y=Y+1
   :POKE PO,P:J=1
27 IF(JOY AND 4)=0 THEN X=X-1
   :POKE PO,P:J=1
28 IF(JOY AND 8)=0 THEN X=X+1
   :POKE PO,P:J=1
29 POKE 55296+X+Y*40,0
   :IF J=1 THEN P=PEEK
   (1024+X+Y*40):J=0
30 IF P>127 THEN P1=P-128
32 IF P<128 THEN P1=P+128
33 PO=1024+X+Y*40
34 REM END CURSOR ROUTINE
35 IF PEEK(53272)=31 THEN CS=2048
   :REM LOWER CASE CHRSET
37 IF PEEK(53272)=29 THEN CS=0
   :REM UPPER CASE CHRSET
41 IF(JOY AND 16)=0 THEN P1=P
   :GOTO 100:REM FIRE BUTTON
42 GET G$:IF G$="S"THEN SYS 49152
43 IF G$="L"THEN LOAD"CHARACTER
   SET1",8,1:REM CHANGE LINE 340
   IF YOU CHANGE THIS!
51 GOTO 20
95 REM MAKE GRID
100 PRINT"[CLEAR]"A$;"[CMDR A,
    CMDR R8,CMDR S]":FOR T=1 TO 8
    :PRINT B$"[CMDR Q,SHFT +8,
    CMDR W]":NEXT
105 PRINT B$"[CMDR Z,CMDR E8,
    CMDR X]"
107 PRINT"[DOWN] [F 7] TO RETURN"
110 N=12288+(8*P)+CS
120 PRINT"[HOME,DOWN,RIGHT]";A$;
    :FOR T=0 TO 7:FOR M=7 TO 0
    STEP-1
121 IF PEEK(N+T)AND 2^M THEN
    PRINT"[RVS] [RVOFF]";
    :GOTO 130:REM DOTS ON
125 PRINT"[SHFT +]";:NEXT
    :GOTO 140:REM DOTS OFF
130 NEXT
140 PRINT:PRINT B$"[RIGHT]";:NEXT
145 X=13:Y=11
148 REM BEGIN GRID CURSOR
    CONTROLL
150 JOY=PEEK(56320)
160 POKE 1024+X+Y*40,P
170 IF(JOY AND 1)=0 THEN Y=Y-1
180 IF(JOY AND 2)=0 THEN Y=Y+1
190 IF(JOY AND 4)=0 THEN X=X-1
195 IF(JOY AND 8)=0 THEN X=X+1
196 GET G$:IF G$="[F7]"THEN CLR
    :GOTO 9
200 P=PEEK(1024+X+Y*40)
210 POKE 1024+X+Y*40,86
    :POKE 55296+X+Y*40,0
220 IF(JOY AND 16)=0 THEN GOSUB
    600:REM FIREBUTTON
```

```
300 GOTO 150
305 REM MACHINE CODE POKED IN
    THRU DATA
310 DATA 169,8,162,8,160,255,32,
    186,255,169,17,162,36,160,
    192,32,189,255,169,0
320 DATA 133,251,169,48,133,252,
    169,251,162,240,160,63,32,
    216,255,96
340 DATA 64,48,58,67,72,65,82,65,
    67,84,69,82,32,83,69,84,49
    :REM @0:CHARACTER SET1
350 REM CHANGING LAST # IN 340
    TO 50 WILL SPELL 'CHARACTER
    SET2', 51 WILL
351 REM MAKE IT 'CHARACTER SET3',
     ETC.
352 REM 4TH # IN DATA LINE 310
    IS DEVICE #. 1=TAPE, 8=DISK
353 REM CHANGE LINE 43 IF YOU
    CHANGE ONE OF THESE DATA
    LINES.
390 IF Q=1 GOTO 12
400 FOR NE=49152 TO 49204:READ XT
    :POKE NE,XT:NEXT
420 RETURN
510 REM END CURSOR
600 IF P=91 THEN P=160:GOTO 630
    :REM CHECK FOR DOT ON
610 IF P=160 THEN P=91:GOTO 680
    :REM CHECK FOR DOT OFF
620 RETURN
625 REM EDIT CHARACTER
630 POKE 12288+(8*P1+CS)+(Y-8),
    PEEK(12288+(8*P1+CS)+(Y-8))
    OR(2^(7-(X-15))):RETURN
680 OP=PEEK(12288+(8*P1+CS)+(Y-8)
    )
690 POKE 12288+(8*P1+CS)+(Y-8),
    OP-(2^(7-(X-15))):RETURN
900 PRINT"[CLEAR,DOWN,SPACE10]
    *** PLEASE WAIT *** "
902 POKE 56334,PEEK(56334)AND 254
903 POKE 1,PEEK(1)AND 251
904 FOR I=0 TO 4096
    :POKE I+12288,PEEK(I+53248)
    :NEXT
905 POKE 1,PEEK(1)OR 4
906 POKE 56334,PEEK(56334)OR 1
907 GOTO 8
998 :
999 :
1000 REM  TO CHOSE A CHAR FOR
     EDITING:
1001 REM  1. POSITION "CURSOR"
     OVER
1002 REM     THE DESIRED LETTER
     USING
1003 REM     THE JOYSTICK.
1004 REM  2. HIT THE FIREBUTTON.
1005 :
1006 REM  TO EDIT A CHARACTER:
1007 REM  1. MOVE THE "CURSOR"
     INTO THE
1008 REM     GRID, USING
     JOYSTICK.
1009 REM  2. PLACE "CURSOR OVER
     DOT
1010 REM     AND HIT FIREBUTTON
     TO
1011 REM     ERASE DOT.
1012 REM  3. PLACE OVER EMPTY
     MATRIX
1013 REM     AND HIT BUTTON TO
     SET DOT.
1014 :
1015 REM  SAVE A CHARACTER SET BY
1016 REM  HITTING "S".
1017 REM  LOAD BY HITTING "L".
1018 :
1019 REM  RETURN TO THE
     CHARACTER MENU
1020 REM  BY HITTING THE
     FUNCTION 7 KEY
1021 :
1022 REM ESCAPE PROGRAM WITH
     RUN/STOP.
```

C

CHARACTERS

Letters to my Grandchildren

Part 1: Roman-to-Arabic and Back

by FH Shedd

Dear Miriam and Lucas:

Last week I began to think about the fact that you two were still using the VIC 20 both at home and at school and that, after all my promises, I had never yet sent you a program. So I started to think about writing one that would be unusual and interesting and—nasty word—*educational*, too. The result is included with this letter and it came about in this way.

Your grandma likes to work crossword puzzles and she had a three-letter word that had a definition "double six". The word turned out to be "XII." And so I got to wondering whether I could easily write a program that would translate Roman numerals to Arabic and Arabic to Roman numerals. I tried it and I'm going to test it on my VIC 20. I think it's short enough for it to handle. We'll see.

Probably both of you have been introduced to Roman numerals in school. However, I'll review them so that you will be able to analyze the program.

You may not have noticed how much they are still used. Whenever a motion picture comes on television there is a "lead in." The major items covered are: title of picture; author; cast; camera and makeup personnel; musicians; etc. And, usually last, there is the year of release, normally given in Roman Numerals. For instance an oldie would show "MCMDLIV", which would mean 1954.

Also look at the inscriptions on monuments, the corner stones of buildings, the large clocks on churches or other public structures and old fashioned time pieces including sundials. Even as late as my boyhood no gentleman or lady would carry a watch with Arabic numbers on the dial.

There was a reason for this, which went back to the Middle Ages. The Arabic numbers were introduced into Europe by contact with the Moors in Spain and the Saracens of Palestine, who were much more advanced in mathematical science—and mainly due to the use of Arabic, or, as the Arabs themselves called them, Hindi numerals.

It was al Khurwarizimi's book, Algebara wal Muquabalah, written in the 9th Century which laid the foundation for mathematical processes using the nine Hindi-Arabic numerals and the zero. From his name comes our word "algorithm" and from the title of the book our word "algebra."

Addition, subtraction, multiplication, division and fractions—in fact all the operations of arithmetic—were greatly simplified by their use. Both mathematicians and merchants caught on to this quickly and began to keep two sets of books. One for their own work records in Arabic numerals—but another to show the public in Roman numerals. It was dangerous to use this "heathen" method and it had to be done in secret. One could actually be accused of being an heretic if caught. And, of course, no public inscription could be made in other than the accepted classical form.

The Semitic languages and Greek used alphabetic characters to represent numbers. This meant that names of people and things could also be read as numbers and, conversely, a certain number also could be translated to a name. It gave rise to whole systems of esoteric properties of numbers. See Numerology, the Cabala, Gematria, Pythagoreanism, The Book of Daniel, The Revelation of Saint John the Divine, etc.

I feel that I am digressing—but at least some knowledge of these uses of numbers is necessary if you want to understand many of the world's religions, Oriental as well as Occidental, and should be part of your background if you are to be an intelligent reader.

It is important that you learn to handle Roman numerals, although today they are little used. One of the tests that can be used to divide the literate from the illiterate, the savant from the idiot, the intellectual from the dolt and the student who makes a higher SAT score from one with a lower is a test of their ability to evaluate Roman numerals. You would be surprised how often some personnel interviewer will pass out a seemingly innocent questionnaire to a job applicant that includes tests of general knowledge.

Anyway, back to mathematics. The basic Roman numbers are:

I	is	1
V	is	5
X	is	10
L	is	50
C	is	100
M	is	1000

If a lower value follows a higher value the count is added:

VI	is	6
XV	is	15
CLX	is	160
MMMD	is	3500

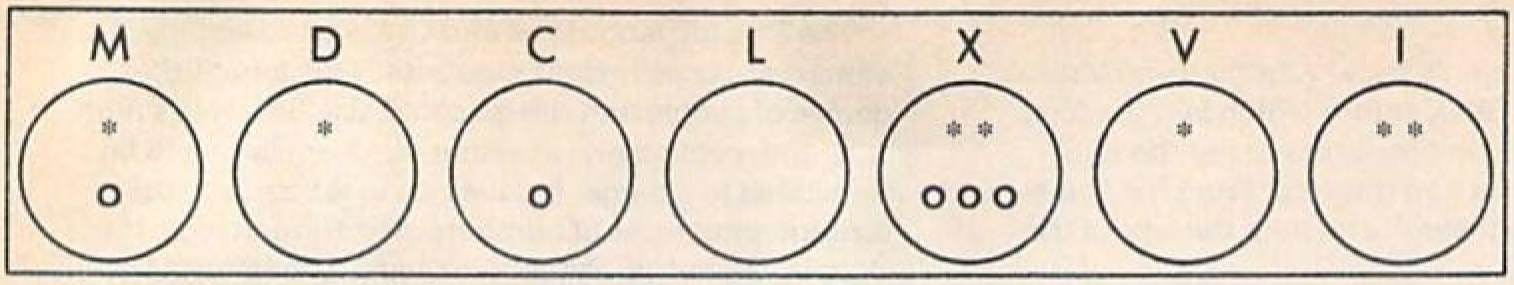
The numbers may be written in a short or long form. The short form to facilitate their writing, the long to facilitate computation. The rule for the short form is: if a lesser value precedes a higher value, the count is subtracted —only one occurrence is allowed and V is not used:

Long Form	Short Form	Is
IIII	IV	4
VIIII	IX	9
MDCCCC	MCM	1900
VIII	IIX	not permitted
LXXXXV	XCV	95
LXXXXV	VC	not used

Now it is obvious that you can't easily add Roman numerals using pencil and paper. Of course the Romans did not have pencils and paper but they did have styli (scratch awls) and tabulae (wax coated tablets). Arabic numbers have fixed positional values and Roman numerals do not. Compare:

	MDXXVII	1529
ADD	MCXXX	1130
	MMDCLVII	2657

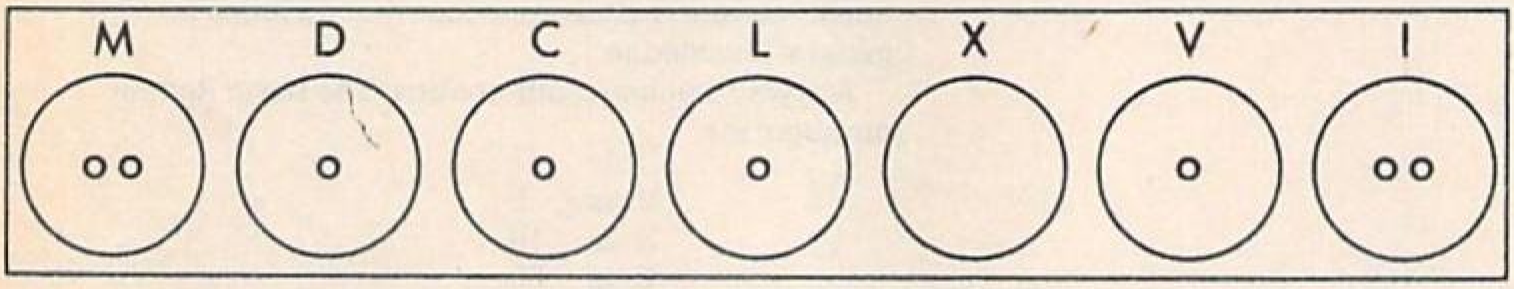
So the Romans had to "calculate." Calculus means "stone" in Latin, their language. They constructed a simple *computer* or *calculator.* They took a handful of small stones and laid out a board with small hollowed out dishes like this:



Observe there is no dish for zero, since they did not recognize this term. The word "zero" comes into our language through an Italian corruption of the Arabic word for empty—sifr—meaning an empty dish on the calculating table.

Stones for the first value, MDXXVII, I have indicated by the symbol "*" and for the second value, MCXXX, the letter "o" in order to identify the individual values. The total is evident but needs simplification.

The rectified table would have the following appearance to indicate the total of 2657 in the short form of MMDCLVII. It is the result of removing the five stones in the X dish and putting one, their equivalent value, in the L dish.



The program is enclosed with "ARABIC-ROMAN" as the file name. It will run on the unexpanded VIC 20. Try it out and let's hear how it works and whether you and your friends find it of interest. I have also enclosed a listing.

When you write be sure to include any questions that may come up when you study this subject. I will try to answer them if I can.

Your grandpa.

Roman-Arabic

```
100  REM PROGRAM NAME =
    ''ROMAN-ARABIC''
110 REM PROGRAM FOR MIRIAM AND
    LUCAS
120 REM      BY GRANDPA 10/7/83
130 PRINT CHR$(147)
140 PRINT "[SPACE7]MENU"
150 PRINT "[SPACE6]=====":PRINT
160 PRINT "1 = ROMAN TO ARABIC"
170 PRINT "2 = ARABIC TO ROMAN"
180 PRINT "3 = STOP PROGRAM"
190 INPUT Z
200 IF Z = 1 THEN GOTO 230
210 IF Z = 2 THEN GOTO 500
220 STOP
230 PRINT CHR$(147)
    :PRINT "ROMAN TO ARABIC"
    :PRINT "0 - RETURN TO MENU"
    :PRINT
240 INPUT "ROMAN # = "; RN$
250 IF RN$="0" THEN GOTO 130
260 L=LEN (RN$)
270 FOR N = 1 TO L
280 L$(N) = MID$(RN$,N,1)
290 GOSUB 420
300 NEXT N
310 S=0:LT=0
320 FOR N=L TO 1 STEP-1
330 T=VAL(P$(N))
340 IF T>=LT THEN S = S +V(N)
350 IF T<LT THEN S = S-V(N)
360 IF T>LT THEN LT=T
370 NEXT
380 PRINT "ARABIC # =";S
```

```
390 PRINT
400 GOTO 240
410 STOP
420 IF L$(N)="M" THEN P$(N)="7"
    :V(N)=1000
430 IF L$(N)="D" THEN P$(N)="6"
    :V(N)=500
440 IF L$(N)="C" THEN P$(N)="5"
    :V(N)=100
450 IF L$(N)="L" THEN P$(N)="4"
    :V(N)=50
460 IF L$(N)="X" THEN P$(N)="3"
    :V(N)=10
470 IF L$(N)="V" THEN P$(N)="2"
    :V(N)=5
480 IF L$(N)="I" THEN P$(N)="1"
    :V(N)=1
490 RETURN
500 PRINT CHR$(147)
    :PRINT "ARABIC TO ROMAN"
    :PRINT "0 - RETURN MENU"
    :PRINT
510 INPUT "ARABIC # = ";AN$
520  IF AN$="0" THEN GOTO 130
530 V=VAL(AN$):IF V < 4999 THEN
    GOTO 560
540 PRINT "4999 IS UPPER LIMIT
    FOR ENTRY"
550 PRINT "GOTO RE-RUN"
    :FOR T=1 TO 1000:NEXT
    :GOTO 130
560 REM R1$="":R2$="":R3$=""
    :R4$=""
570 L = LEN (AN$)
580 FOR N = 1 TO L
590 M$(N)=MID$(AN$,N,1)
600 NEXT N
610 R4$="":R3$="":R2$="":R1$=""
620 IF L=1 THEN GOSUB 1120
630 IF L=2 THEN GOSUB 1160
640 IF L=3 THEN GOSUB 1200
650 IF L=4 THEN GOSUB 1250
660 PR$= R4$+R3$+R2$+R1$
670  L=LEN(PR$):IF L<10 THEN
    PR$=" "+ PR$:GOTO 670
680 PRINT "ROMAN = ";
690 PRINT PR$
700 PR$="[SPACE11]"
710 R4$="":R3$="":R2$="":R1$=""
720 PRINT
730 GOTO 510
740 IF I$= "1" THEN R1$= "I"
750 IF I$= "2" THEN R1$= "II"
760 IF I$= "3" THEN R1$= "III"
770 IF I$= "4" THEN R1$= "IV"
```

```
780 IF I$= "5" THEN R1$= "V"
790 IF I$= "6" THEN R1$= "VI
800 IF I$= "7" THEN R1$= "VII"
810 IF I$= "8" THEN R1$= "VIII"
820 IF I$= "9" THEN R1$= "IX"
830 IF J$= "1" THEN R2$= "X"
840 IF J$= "2" THEN R2$= "XX"
850 IF J$= "3" THEN R2$= "XXX"
860 IF J$= "4" THEN R2$= "XL"
870 IF J$= "5" THEN R2$= "L"
880 IF J$= "6" THEN R2$= "LX"
890 IF J$= "7" THEN R2$= "LXX"
900 IF J$= "8" THEN R2$= "LXXX"
910 IF J$= "9" THEN R2$= "XC"
920 IF K$= "1" THEN R3$= "C"
930 IF K$= "2" THEN R3$= "CC"
940 IF K$= "3" THEN R3$= "CCC"
950 IF K$= "4" THEN R3$= "CD"
960 IF K$= "5" THEN R3$= "D"
970 IF K$= "6" THEN R3$= "DC"
980 IF K$= "7" THEN R3$= "DCC"
990 IF K$= "8" THEN R3$= "DCCC"
1000 IF K$= "9" THEN R3$= "CM"
1010 IF L$= "5" THEN R4$= "L"
1020 IF L$= "1" THEN R4$= "M"
1030 IF L$= "2" THEN R4$= "MM"
1040 IF L$= "3" THEN R4$= "MMM"
1050 IF L$= "4" THEN R4$= "MMMM"
1060 IF L$= "5" THEN R4$= "L"
1070 IF L$= "6" THEN R4$= "LX"
1080 IF L$= "7" THEN R4$= "LXX"
1090 IF L$= "8" THEN R4$= "LXXX"
1100 IF L$= "9" THEN R4$= "CX"
1110 RETURN
1120 I$=M$(1)
1130 GOSUB 740
1140 R4$="":R3$="":R2$=""
1150 RETURN
1160 J$=M$(1):I$=M$(2):GOSUB 740
1170 IF M$(2)="0" THEN R1$=""
1180 R4$="":R3$=""
1190 RETURN
1200 K$=M$(1):J$=M$(2):I$=M$(3)
     :GOSUB 740
1210 IF M$(2)="0" THEN R2$=""
1220 IF M$(3)="0" THEN R1$=""
1230 R4$=""
1240 RETURN
1250 L$=M$(1):K$=M$(2):J$=M$(3)
     :I$=M$(4):GOSUB 740
1260 IF M$(2)="0" THEN R3$=""
1270 IF M$(3)="0" THEN R2$=""
1280 IF M$(4)="0" THEN R1$=""
1290 RETURN
```

C

Automatic VIC 20 Disk Starter

by Stephen S. Leven

For over a year I've had a VIC 20, and I have been accumulating, among other things, a large collection of game programs on tape for my kids. Many of these came from typing the listings found in magazines like *Power/Play,* and some came from friends and other members of users groups. The main game player in our household is my 6 year old son, Jonathan, so I SAVED each game on a separate tape. Jonathan is just beginning to learn to read, so I used different colors or sketches on the label of each tape to help him identify the games he wants to play.

It was a simple task to teach him to turn on the VIC and LOAD the games himself. All he had to do was (1) switch on the VIC and the TV, (2) select the game he wanted, (3) insert the cassette into the datassette, (4) hold down the SHIFT key and press the RUN/STOP key, and (5) press PLAY on the datassette. Jonathan got very adept at this procedure, which is remarkable for a child who sometimes can't remember whether his shoes or socks go on first!

However, I noticed that sometimes in the 40 to 60 seconds that some games took to LOAD, Jonathan would grow impatient. This was just one more incentive for me to buy the 1541 disk drive. Now, neither of us would get impatient!

A Mixed Blessing

After connecting the 1541 and reading the manual, I began playing with the VIC and disk drive. I soon realized that even though programs LOADed much more quickly than with tape, the system was a bit more complex for a child (or adult, for that matter) to use than the datassette. It wasn't hard to teach Jonathan to turn the disk drive on before turning on the VIC 20. It also wasn't hard to teach him the proper way to handle the disks, insert them into the drive and remove them and store them after LOADing the program. What was difficult was to teach him to enter the commands necessary to set the drive to the proper speed for the VIC and to LOAD the program of his choice.

As specified in the manual for the disk drive, the following command must be used when the drive is first turned on to set the speed of data transfer to match the VIC. (When powered up, the 1541 is set at a slightly slower speed to match the Commodore 64):

```
OPEN 15,8,15,"UI-" : CLOSE 15[RETURN]
```

Since I had SAVEd all the games on one disk, titled GAMES, the commands to LOAD and RUN a particular game are as follows:

```
LOAD "game name",8   [RETURN]
RUN                  [RETURN]
```

This is a lot for a non-reading, non-writing (and non-typing) child to enter into a computer. Jonathan seemed disappointed that the much heralded disk drive was not so easy to use, from his point of view, after all.

Enter the Menu

As an owner of a Commodore 64 computer, too, (like my wife, I just can't resist a bargain, and the 64 really is a bargain!) I had a copy of the Disk Bonus Pack. This disk, which is completely filled with useful utility, game and demonstration programs for the 64, included a menu program, C64. MENU, written by Michael Schaff. If the menu program is SAVEd as the first program on the disk, it can be LOADed and RUN by simply typing the following command:

```
LOAD ":*",8 : [SHIFT-RUN/STOP] [RETURN]
```

However, as you might expect, when I tried to run C64. MENU on the VIC, it didn't work. Fortunately, the program is in BASIC, so I listed it and set about changing it to work on the VIC 20. Program 1 at the end of this article is the menu program, modified to work on the VIC 20, with any amount of memory.

When VIC MENU is RUN, the first thing it does is search the disk for a file titled "DIRECTORY". If it finds this file, it reads the contents and displays a listing of the first 15 program files in alphabetical order. Next to each program name is a number. If you wish to LOAD and RUN any program displayed on the menu, simply enter the number of the program, and press RETURN. If there are more than 15 programs on the disk, pressing F1 will display the next "page" of 15 programs, numbered 16 through 30. This process will continue until all of the programs have been displayed. Pressing F1 again returns you to "page 1", and the process begins again.

If there is no file titled DIRECTORY, the program creates one. You will hear the disk drive stop spinning with the red light still on one or more times during this operation. This is normal. Also, if you have added new files to the disk since the time the DIRECTORY file was created, you can press F3 when the first "page" is displayed, and a new DIRECTORY file will be created.

Pressing F5 ends the program, but leaves you with the cursor on a line on which the word RUN is printed. If you want to use VIC MENU, which is still in your computer's memory, on another disk (which need not have the VIC MENU program on it), simply place the new disk in the drive and hit RETURN. The program will RUN, and the directory of the new disk will be displayed.

The VIC MENU displays only program files, but not any sequential or other types of files. In addition, any program name that starts with a space is not displayed on the menu. This can be helpful when one program LOADs another. More on this later.

I SAVEd VIC MENU as the first program on the GAMES disk (See page 110 for the procedure on how to do this to a disk with programs already on it). Now, Jonathan could simply (??) type:

```
      OPEN 15,8,15,"UI-" : CLOSE 15   [RETURN]
    LOAD ":*",8 : [SHIFT-RUN/STOP]   [RETURN]
```

and then he could select the game by number from the menu. I even used some of the VIC graphic symbols as part of the name of the game program when I SAVEd it, so he would be better able to identify the game he wanted. (Yes, graphic characters can be used in program names, just like letters and numbers.)

The Disk Starter

This was still a lot of typing for a youngster (it's a lot of typing for me!) so I thought of a way to further simplify the whole process. What if I could get the VIC to do this typing for me, by using the dynamic keyboard technique? What if I put a short program on tape, which can be LOADED and RUN by the simple press of the SHIFT and the RUN/STOP keys, and which would automatically set the disk drive speed and LOAD and RUN the menu? That's what the DISK STARTER is all about.

Program 2 is the VIC DISK STARTER. This short program prints the OPEN/CLOSE command and the LOAD/RUN command on the screen, in the same color as the background so you don't see it. Then, the keyboard buffer is filled with the proper number of HOME, CRSR DOWN, and RETURN characters to execute these statements. And when SAVEd to tape, the DISK STARTER can be activated with the simple SHIFT-RUN/STOP key sequence. No more typing!!

Putting it All Together

Here's what to do. First, format a new disk. The manual that came with your 1541 disk drive tells you how to do this. Next, type in Program 1, the VIC MENU. SAVE the VIC MENU as the first program on your newly formatted disk. Be sure to leave a space between the first quote mark and the letter V, as shown below:

```
          SAVE "[SPACE]VIC MENU",8
```

Now LOAD your first game program (or any program you wish to use this method for), then SAVE it on the disk with the VIC MENU. Repeat this process until all the programs you want are on the disk. You can make up multiple disks, such as one for games, one for home finance programs, one for utilities, etc. Just be sure that the VIC MENU is the first program on the disk.

Next, type in Program 2, the VIC DISK STARTER. Be sure to leave a space between the third quote mark and the letter V on line 20. This must match the name of the program you just SAVEd on disk, character for character. SAVE this program on tape!!. SAVE it just twice until you're sure it's working properly. Then, put a fresh tape on the datassette and enter the following command in direct mode:

```
FOR I=1TO100 : SAVE "VIC DISK STARTER" :
                        NEXT [RETURN]
```

The program will be SAVED over and over again on the same tape, until the tape stops. (Actually, one side of a 60-minute tape, which takes 30 minutes, will hold about 97 copies of the VIC DISK STARTER). When the tape stops, don't rewind it, but turn it over, hold down RUN/STOP and press RESTORE, and repeat the process. You now have VIC DISK STARTER SAVEd continuously on both sides of the tape! Now, rewind the tape to about the middle.

Let's Put it to Work

Now we have a system that even Jonathan can use. Here's the procedure for LOADing and RUNning any program from disk on the VIC 20, with minimum typing effort:

1. Turn on the monitor or TV, the disk drive, and then the VIC 20.
2. Place the disk containing the program you want to RUN and the VIC MENU program in the disk drive.
3. Place the DISK STARTER tape in the datassette, either side up.
4. Hold down the SHIFT key and press RUN/STOP.
5. The VIC 20 will say "PRESS PLAY ON TAPE". Do it.

The tape will begin moving in the datassette. Soon the VIC will display "FOUND VIC DISK STARTER", then "LOADING". After a few seconds, the VIC screen will clear, and display the title "VIC MENU". Seconds later, the menu of programs on the disk will appear. Simply find the program of interest (use the F1 key to view the next "page", if necessary), type in its number, press RETURN and the program will LOAD and RUN automatically. Even I can manage that procedure!

Helpful Hints

IMPORTANT: Be sure the title of the VIC MENU program you SAVE to disk begins with a SPACE! Be sure that the title of the VIC MENU program you type in line 20 of the VIC DISK STARTER also begins with a SPACE. These two program names must match, character for character, or the VIC DISK STARTER will not find the VIC MENU on the disk.

There is never any need to rewind the VIC DISK STARTER tape. Simply place it in the datassette with more tape on the left side than the right.

Keep the PLAY button depressed on the datassette until your session with the VIC is finished. That way, on subsequent start-ups, you won't have to press this button. DON'T FORGET TO PRESS STOP AND REMOVE THE TAPE WHEN YOU ARE DONE WITH YOUR SESSION ON THE VIC 20!!

Don't forget to remove the disk from the drive before turning off power to either the computer or the disk drive. Turn off power to the VIC 20 before turning off power to the 1541 drive.

Program your games, and other programs, with a definite end. A statement such as "PLAY AGAIN? Y/N : " in a game program or a menu item "9. END PROGRAM" will do the trick. Then program the END or NO choice to re-LOAD and RUN the VIC MENU. Program lines such as these will do the trick:

```
900 PRINT"[CLEAR]PLAY AGAIN? Y
    OR N:"
910 GET A$:IF A$="Y"THEN 000
    : REM 000 IS THE BEGINNING
    OF THE PROGRAM, AFTER INIT.
920 IF A$<>"N"THEN 910
930 PRINT"[CLEAR]LOAD"CHR$(34)"
    VIC MENU"CHR$(34);",8[DOWN10]
    ":PRINT"ONE MOMENT PLEASE..."
940 POKE 631,19:POKE 632,13
    :POKE 633,82:POKE 634,117
    :POKE 635,13:POKE 198,5:END
```

These lines, in the appropriate part of the program (change the line numbers as necessary), will clear the screen and cause the question "PLAY AGAIN? Y OR N :" to be printed at the top of the screen. Pressing the Y key will cause the program to begin again. Pressing the N key will cause VIC MENU to be LOADed and RUN. Pressing any other key (except RUN/STOP) will have no effect.

The fact that the VIC MENU doesn't display program names that begin with a SPACE, provides a convenient method for using a series of programs, that is, one program which LOADs and RUNs another. For example, suppose you're entering a two-part program named GAME. Part 1 is the instruction screen, custom character generator and initialization. Part 2 is the actual game itself. SAVE Part 1 in the normal manner:

SAVE "GAME",8

Now, SAVE Part 2 beginning with a SPACE:

SAVE "[SPACE]GAME PART 2",8

Be sure that the Part 2 LOAD command in Part 1 includes the SPACE as part of the program name. Now, when VIC MENU is RUN, only the first part, titled GAME, will appear in the menu. This will prevent you from entering the item number for the wrong part. When you LIST the disk directory, using either the normal method or the DOS WEDGE, both parts of GAME will show up, and the name of GAME PART 2 will begin with a blank space.

Typing Help

If you don't want to do all the typing yourself, just send me a blank formatted (with the NEW command) disk (no tapes please!), a self-addressed stamped return mailer and $3.00, and I'll be glad to make you copies of both programs. When you get your disk back, just LOAD the VIC DISK STARTER into your VIC 20, then SAVE it 97 times or so on both sides of a blank tape. My address is:

6 Hillery Court
Randallstown, MD 21133

Besides being a great game machine, the VIC 20 is a real computer! We might as well let it do some of the tedious work, like repetitive typing, for us. Making life easier is what the computer age is all about.

Adding VIC MENU as the First Program on an Existing Disk

If you have a disk full of programs already and want to add the VIC MENU as the first program on this disk, you must move the existing first program to elsewhere on the disk, then SAVE VIC MENU. The following procedure must be used. (Let's assume the first program on your existing disk is named GAME I):

1. If you haven't already done so, type:
   OPEN 15,8,15,"UI-" [RETURN]
   This opens the disk command channel and sets the 1541 disk drive to the faster speed, compatible with the VIC 20.
2. Rename your first program to a temporary name (file 15 must be OPEN for the disk command channel)
   PRINT#15, "RO: TEMPORARY=GAME 1" [RETURN]
3. Copy the file from TEMPORARY to another file with the original name. You will now have two copies of the file on the disk, with two different names:
   PRINT#15, "CO: GAME 1=TEMPORARY" [RETURN]
4. Scratch the first file on the disk, which is the one named TEMPORARY:
   PRINT#15, "SO: TEMPORARY" [RETURN]
5. SAVE the VIC MENU. It will be SAVEd in the first available space on the disk, which is now the first directory item:
   SAVE"[SPACE]VIC MENU",8 [RETURN]

Program 1. VIC MENU

```
5 FL=0:SM=8185:CM=38905
  :IF PEEK(56)>30 THEN SM=4601
  :CM=38393
10 POKE 36879,110:DIM TB$(150)
   :PRINT"[CLEAR]":DI=8:C=1
15 PRINT"[DOWN4]"SPC(6)"[WHITE]
   VIC MENU[BLUE]"
20 OPEN 8,DI,8,":[SPACE3]
   DIRECTORY[SPACE3]"
   :OPEN 15,DI,15:INPUT#15,EN$,
   ED$,ET$,ES$
30 INPUT#8,DN$:IF EN$<>"00"THEN
   CLOSE 8:CLOSE 15:GOTO 310
40 INPUT#8,TB$(C):IF ST=66 THEN
   70
50 IF ASC(LEFT$(TB$(C),
   1))=10 THEN TB$(C)=RIGHT$(TB$
   (C),LEN(TB$(C))-1):GOTO 50
60 C=C+1:GOTO 40
70 CLOSE 8:CLOSE 15:M=C-1
   :TB$(C)="":L1$="[RVS]"
   :FOR C=1 TO 22:L1$=L1$+" "
   :NEXT
80 L1$=L1$+"[RVOFF]":A=1
```

```
90 POKE 36879,30:PRINT"[CLEAR,
   BLUE]";:FOR C=1 TO 4
   :PRINT L1$;:NEXT
95 PRINT"[HOME]";
100 PRINT SPC(1)"[RVS,BLUE]
    DIRECTORY ASSISTANCE"SPC(31)
    "VIC-20"SPC(30)
120 FOR C=A TO A+14
    :PRINT"[RVS]"RIGHT$("[SPACE2]
    "+STR$(C),3)"[RVOFF] [BLUE]
    "TB$(C)SPC(16-LEN(TB$(C)));"
    [RVS,SPACE2]";
130 NEXT:PRINT L1$;
135 PRINT"[BLUE]"L1$;
    :PRINT LEFT$(L1$,22)"[UP]"
140 PRINT SPC(1)"[RVS,UP2]
    F1 = CONTINUE LISTING"SPC(1)
    "F3 = CREATE DIRECTORY";
150 PRINT"[RVS] F5 = EXIT
    PROGRAM[SPACE3,BLUE,HOME]"
    :POKE SM,160:POKE CM,6
    :POKE 198,0:B$=""
160 GET A$:IF A$=""THEN 160
170 IF ASC(A$)=13 THEN 260
180 IF A$="[F1]"THEN 240
190 IF A$="[F3]"THEN 300
200 IF A$="[F5]"THEN POKE 36879,
    27:PRINT"[CLEAR,DOWN3,BLUE]
    RUN[HOME]":END
210 IF ASC(A$)=20 THEN B$=""
    :PRINT"[HOME,DOWN3,RED,RVS,
    SPACE3,RVOFF]":B$="":GOTO 160
220 IF A$<"0"OR A$>"9"OR
    LEN(B$)=3 THEN 160
230 B$=B$+A$:PRINT"[HOME,DOWN3,
    BLUE]"B$:GOTO 160
240 IF C>M THEN A=1:GOTO 90
250 A=C:GOTO 90
260 IF VAL(B$)>M OR B$=""THEN
    PRINT"[HOME,DOWN3,BLUE]???"
    :FOR C=1 TO 1500:NEXT
    :A$=CHR$(20):GOTO 210
270 PRINT"[RVOFF,CLEAR,DOWN3,
    WHITE]LOAD"CHR$(34)TB$(VAL
    (B$))CHR$(34)",";
280 PRINT RIGHT$(STR$(DI),
    LEN(STR$(DI))-1)
290 PRINT"[HOME,BLUE]":POKE 198,4
    :POKE 631,13:POKE 632,82
    :POKE 633,213:POKE 634,13:END
300 PRINT"[HOME,DOWN3,BLUE]WAIT"
310 OPEN 8,DI,0,"$0":FOR C=1 TO 8
    :GET#8,A$:NEXT:C=1:DN$=""
    :FOR C=1 TO 16
320 GET#8,A$:DN$=DN$+A$:NEXT
    :GET#8,A$:GET#8,A$
    :DN$=DN$+"[SPACE2]":GET#8,A$
330 DN$=DN$+A$:GET#8,A$
    :DN$=DN$+A$:GET#8,A$:GET#8,A$
    :DN$=DN$+"[SPACE2]"+A$
340 GET#8,A$:DN$=DN$+A$:GET#8,A$
    :C=1
350 FOR A=1 TO 4:GET#8,A$:NEXT
    :PN$="":TY$=""
360 GET#8,A$:IF ST<>0 THEN 450
370 IF A$=""THEN 450
380 IF ASC(A$)<>34 THEN 360
390 GET#8,A$:IF ASC(A$)<>34 THEN
    PN$=PN$+A$:GOTO 390
400 GET#8,A$:IF ASC(A$)=32 THEN
    400
410 TY$=TY$+A$:GET#8,A$
    :IF A$<>""THEN 410
420 IF LEFT$(TY$,3)<>"PRG"THEN
    350
430 IF LEFT$(PN$,1)=" "THEN 350
440 TB$(C)=PN$:C=C+1
    :IF ST=0 THEN 350
450 CLOSE 8:OPEN 15,DI,15,"S
    :[SPACE3]DIRECTORY[SPACE3]"
    :CLOSE 15
460 Z$="[SHFT Z16]":OPEN 8,DI,8,"
    :[SPACE3]DIRECTORY[SPACE3],S,
    W":PRINT#8,DN$
470 FOR A=1 TO C-1:C$=Z$
    :FOR B=1 TO C-1
    :IF C$<TB$(B)THEN 490
480 C$=TB$(B):D=B
490 NEXT:PRINT#8,C$:TB$(D)=Z$
    :NEXT:CLOSE 8:CLR:GOTO 5
```

Program 2. VIC Disk Starter

```
10 PRINT"[CLEAR,WHITE]RUN"
20 PRINT"LOAD"CHR$(34)" VIC
   MENU"CHR$(34)",8
30 PRINT"OPEN15,8,15,
   "CHR$(34)"UI-"CHR$(34)"
   :CLOSE15"
40 POKE 631,19:POKE 632,17
   :POKE 633,17:POKE 634,13
   :POKE 635,19:POKE 636,17
50 POKE 637,13:POKE 638,19
   :POKE 639,13:POKE 640,31
   :POKE 198,10
```

C

All You Need to Know to Begin Programming in Machine Language

Part 1

by John J. Karcher

This article is a compilation of information from various sources—all the information needed to begin programming in 6502 machine language on the VIC 20. The object of this article is to save you the effort necessary to gather and organize all this information, and to explain it in understandable terms. This article is divided into several short sections, each a different aspect of machine language programming.

When I first tried to learn machine language, all the information I read said that to learn machine language one must understand how the VIC works. But I did not believe this. However, as I progressed, I learned that a fundamental knowledge of the VIC's internal functions is very helpful. This is by far the hardest part of learning machine language. The rest is very simple.

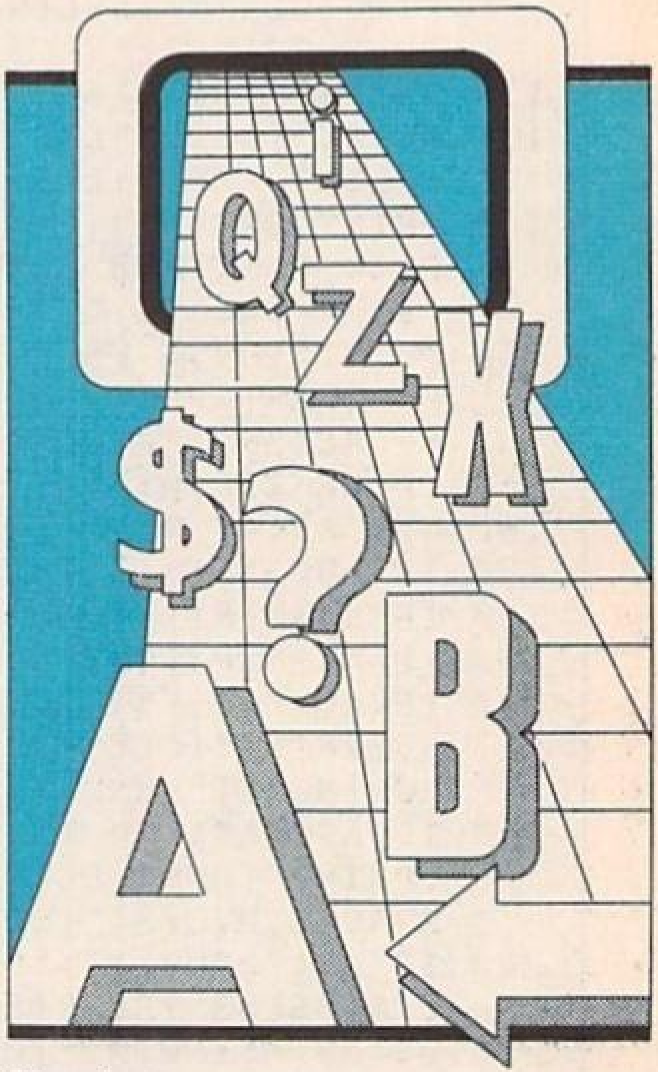
Introduction To Machine Language

As you know, the VIC's memory is made up of locations, each containing a number between zero and 255. A question arises from this fact. If the VIC stores only numbers, how does it store our BASIC programs? Well, the VIC assigns each BASIC command and every character a number, (between zero and 255, of course). A list of these values is on page 121 of the *VIC 20 Programmer's Reference Guide*. The VIC automatically stores these values in memory, between locations 4096 and 7679 and shuffles them around when you edit your programs.

The BASIC language is a machine language program, which tells the VIC what to do when it sees the program in memory. If BASIC is a machine language program, is machine language a program in another, super, language? No, machine language is the actual language of the VIC's microprocessor. That's right! Machine language is as close as you can get to the internal workings of the VIC.

Since machine language is the microprocessor's language, it has many advantages over BASIC. First, machine language is very, very fast. In fact, it can be *hundreds* of times faster than BASIC. Second, it lets only what you want to happen, happen. The user cannot destroy your program if you program it to be user-safe. Third, it lets you do more things than BASIC. Fourth, it uses much less memory to do all of this.

But machine language has its disadvantages too. While no error messages tell you about mistakes, an error causes the VIC to "hang up", or, in other words, all signs of usability disappear. You will have to turn the VIC off then on to regain control. But, as you may have guessed, when you do this, you lose your program, just as you do with BASIC. Also, each command does only one thing. But the disadvantages are outweighed by the advantages.



Numbers

In programming, three number bases are used—two, ten and sixteen. What is a number base? I'll explain. When, in the early grades of school, the teachers talked about numbers, they spoke of the one's column, the ten's column, and the hundred's column. For example, 126 is said to have one "hundred", two "tens", and six "ones". This is base ten, because each column has *ten* times the significance of the one on its right.

What about base two? Each digit has *two* times the significance of the one to its right. For example, 101 in binary (base two) is one "four", zero "two", and one "one".

The highest one-digit counting number in a base is one less than the base number. Let's count to 16 in base 16—one, two, three, four, five, six, seven, eight, nine, ?!? What next? The next number cannot be more than one digit long, so ten is written A, 11 is B, 12 is C, etc., up to F for 15. For example, 25B in hexadecimal (base 16) is two "256's" (Remember, each digit has 16 times the significance of the one to its right, 16× 16 = 256.), five "16's", and B (11) "ones". When added together these equal 603 in base ten. The number sixteen is written "10" because there is one "16" and zero "ones".

This is important, so read carefully. In hexadecimal (or any base, for that matter) ten plus 20 still equals 30; two times 30 is still 60. This is true as long as both numbers are in the same base. But 20_{10} times 30_{16} does not equal 600 hexadecimal or decimal. *When you're working in any number system, think only in that system.*

As you can see, all calculations are performed the same way. You don't have to re-learn addition, subtraction, multiplication or division to work in a different number base. In decimal, 2× 8= 16; in hexadecimal, 2× 8= 10 (one "16" and zero "ones")! Also remember, in hexadecimal 19 plus one equals 1A!!! Always keep this kind of thing in mind. Counting incorrectly could be disastrous!

What significance does base 16 have? Binary (base two) is what the VIC actually uses, and decimal (base ten) is what BASIC uses. Hexadecimal (base 16) is a compromise. Can you imagine writing 7680 in binary? It is not bad in decimal, but conversion between bases two and ten is slow, so each four digits of binary were made into one hexademical digit, for the programmer's comfort only.

For example, 0010110101101100 can be divided every four digits—0010,1101,0110,1100. Let's find the values of these groups starting with the first— 0 "eights", 0 "fours", 1 "two", 0 "ones" = 2. Do the others the same way and you get four numbers—2,13,6, and 12. Convert these to hexadecimal notation and you get 2D6C. Which would you rather write twenty times in a program—2D6C or 0010110101101100?

The "address" is the number given to a memory location. Addresses can be from 0000 to FFFF in hexadecimal (0 to 65535 in decimal). Memory is divided into FF pages. The "page number" is the first two digits of the hexadecimal address. For example, 5687_{16} is in page 56. The first address in page 01 is 0100, and the last is 01FF. By the way, the page number does not have anything to do with the values in the page.

Something else to keep in mind —a hexadecimal address must be four digits long, a value must be two. For example, address 0100 must be written 0100, not 100. The value 0F must be written 0F, not just F. An important fact to remember is that 255 in decimal is FF in hexadecimal. This is the highest two-digit number possible, meaning that any two-digit number in hexadecimal, including zeros, will fill a single byte.

Conversion

You have already learned to change binary to hexadecimal, and changing back is just as simple. I will not explain how this works as I go along. You will just have to believe me. It does work. Use a calculator for convenience.

Hexadecimal to decimal: Take a hexadecimal number, like 7B54. Take the first digit, 7, and multiply by 16. Add the second digit, B, and multiply by 16 again. Add the third, 5, and multiply by 16. Finally, add the last digit.

7× 16= 112+ 11= 123× 16= 1968+ 5=
1973× 16= 31568+ 4= 31572
$7B54_{16} = 31572_{10}$

For two-digit numbers it works the same way: C5.

12× 16= 192+ 5= 197
$C5_{16} = 197_{10}$

Decimal to hexadecimal: Now, take a decimal number, this time 31572. First, you divide by 4096. You get a number around 7.708. The first digit is the 7 to the left of the decimal point. Subtract this 7 from the number on the display and multiply by 16. You get a number around 11.328. Eleven (B) is the next digit. Subtract 11 from the number on the display and multiply again by 16. The answer is almost exactly 5.25. Five is the next digit. Subtract 5 and multiply by 16. The final answer is very near 4. The last digit is 4. The hexadecimal equivalent of 31572 is 7B54.

31572÷ 4096= 7.708− 7= .708× 16=
11.328− 11= .328× 16= 5.25− 5=
.25× 16= 4
(7B54) $31572_{10} = 7B54_{16}$

For numbers above 256, use the above method. Here is a short cut for numbers below 256. Take 197, for example. First, divide by 16. You get 12.3125. Twelve is the first digit. Subtract 12 from the number. Multiply by 16. The answer is 5. The hexadecimal equivalent of 197 is C5.

197÷ 16= 12.3125− 12= .3125× 16= 5
(C5) $197_{10} = C5_{16}$

NOTE: *If you get a number like 2.99999 or 10.0001 as a* final answer, *then just round it off.*

Coding

Once you have made it this far, you are home free. Remember when I told you how your VIC stores your programs? It stores machine-language programs the same way. The list of strange-looking commands on pages 164-167 of the *VIC 20 Programmer's Reference Guide* is similar to the one on page 121, except it is for machine language. By the way, it's numbered in hexadecimal. On pages 140 and 141 is a list of commands, and on pages 142-161 is technical information.

At first glance (and second, and third, for that matter), all this looks forbiddingly complicated. You might not be able to understand anything at all. Don't despair, though, it is all fairly simple, once you understand a few things. Here are some guidelines.

One of the first things you'll notice on pages 140 and 141 is that two indexes and the word "accumulator" keep popping up. What is the "accumulator"? It is a memory location —780 in decimal, 030C in hexadecimal. What makes this location special? Nothing, except that the 6502 microprocessor has set it aside for storing special numbers. The X and Y indexes work the same way.

One more piece of information before we go on. Machine language programs are stored directly in memory, using no line numbers. The only reference points, then, are the memory addresses within which the program is stored.

Commands

Each command is stored in memory as a number between 0 and 255 (00 to FF hexadecimal). Your machine-language programs are a series of these numbers. Working only with numbers is fine for short programs, but numbers are far harder to read in program listings than words. So, some early programmers or chip designers, or someone (I don't really know who) developed "assembly language", to make it easier to program using machine language. In assembly language, each command is assigned a three-letter word. To distinguish between modes (I will explain these modes later), syntax is changed a little.

* **ADC**	—Add a number to the value in the accumulator.
* **AND**	—"AND" a number with the accumulator value (this is a logical operation).
ASL	—Shift left one bit (accumulator or other).
BCC	—Branch if carry flag is clear (0).
BCS	—Branch if carry flat is set (1).
BEQ	—Branch if result is zero (0).
BIT	—Test bits in memory against the accumulator.
BMI	—Branch if result is negative.
BNE	—Branch if result does not equal zero (0).
BPL	—Branch if result is positive.
BRK	—Force break; like END or STOP in BASIC.
BVC	—Branch if overflow flag is clear (0).
BVS	—Branch if overflow flag is set (1).
CLC	—Clear carry flag.
CLD	—Clear decimal mode.
CLI	—Clear interrupt disable bit.
CLV	—Clear overflow flag.
* **CMP**	—Compare a number with accumulator.
* **CPX**	—Compare a number and the X index.
* **CPY**	—Compare a number and the Y index.
DEC	—Decrement (subtract one from) a memory location.
DEX	—Decrement the X index.
DEY	—Decrement the Y index.
* **EOR**	—"Exclusive—Or" a number and the accumulator (this is a logical operation).
INC	—Increment (add one to) a memory location.
INX	—Increment the X index.
INY	—Increment the Y index.
JMP	—Jump to a new location; like GOTO in BASIC.
JSR	—Jump to a new location saving return address; like GOSUB in BASIC; see RTS.
* **LDA**	—Load the accumulator with a number.
* **LDX**	—Load the X index with a number.
* **LDY**	—Load the Y index with a number.
LSR	—Shift right one bit (accumulator or other).
NOP	—No operation—time delay.
* **ORA**	—"OR" a number with the accumulator (this is a logical operation).
PHA	—Push accumulator into stack.
PHP	—Push processor status into stack.
PLA	—Pull accumulator from stack.
PLP	—Pull processor status from stack.
ROL	—Rotate one bit left (accumulator or other).
ROR	—Rotate one bit right (accumulator or other).
RTI	—Return from interrupt.
RTS	—Return from subroutine; like RETURN in BASIC; see JSR.
* **SBC**	—Subtract a number from the accumulator.
SEC	—Set carry flag (1).
SED	—Set decimal mode (1).
SEI	—Set interupt disable status (1).
STA	—Store accumulator in memory.
STX	—Store X index in memory.
STY	—Store Y index in memory.
TAX	—Transfer accumulator to X index.
TAY	—Transfer accumulator to Y index.
TSX	—Transfer stack pointer to X index.
TXA	—Transfer X index to accumulator.
TXS	—Transfer X index to stack pointer.
TYA	—Transfer Y index to accumulator.

Not just any combination of letters will do for any command. Whoever developed the 6502 assembly language "mnemonics" (a fancy name for our three-letter words) did a very good job in my opinion. For example, the command that puts a number in X is LDX, probably from LoaD the X register with a value. STX—STores the value in the X register, somewhere in memory. CLC CLears the Carry flag. Because of this, reading a program listing becomes second nature quickly.

If you try using assembly language in your computer, it won't work. To use assembly language, you must use an "assembler". An assembler is a piece of software that allows you to program in assembly language. Most machine language monitors include a mini-assembler. A mini-assembler is an assembler without all the fancy features of a full-featured assembler. But a good monitor will make up for this with *very* helpful editing capabilities.

On this page is a list of commands, and an explanation of each one. If you have the *Programmer's Reference Guide,* this information is on pages 140 and 141. If not, I suggest you get one, but this article will suffice temporarily.

You may have noticed, while glancing down the list, that several commands have stars next to them (the star is not part of the command). You may also have noticed that they all apply to the accumulator, the X index and the Y index. They also frequently have the word "number" in their definitions. (I made sure of this. In the *Programmer's Reference Guide,* this is not the case. This source uses "memory", which can be confusing).

The syntax of assembly language is simple. First, you write the command. For some commands this is all that is needed. But for most (including the ones with stars) you follow the command with a number. For example, LDY "number" or STA "number". This "number" is what the next section is about.

Immediate, Absolute or Zero Page

These are the three modes of which this "number" can be. For those of you who do not understand what these words mean, there is a simple explanation. By the way, the mysterious "$" symbol in our examples indicates our "number" is in hexadecimal, and is *absolutely* necessary, at all times. Also, from now on, all numbering is in hexadecimal.

Immediate—This simply means that the "number" is a legal *value* (a number from 00 to FF), and is *the value* used by the command. Example: ADC #$10. Let's pretend that the value in the accumulator is 20. Twenty plus ten equals thirty. Thirty is returned to the accumulator. The "#" symbol indicates immediate mode.

Absolute—This means that our "number" is a legal *address* (0000-FFFF), and is the *address of the value* to be used by the command. Example: ADC $1C06. The accumulator value is still 20. The value in 1C06 is 15. Twenty plus 15 equals 35. Thirty-five is returned to the accumulator. The lack of the "#" symbol and a four-digit "number" indicate absolute mode.

Zero Page—This is very similar to absolute, except that the first two digits of our "number" are zeros (The address is in page zero.). Thus, they are omitted. Example: ADC $C5. The accumulator value is 20 again. The value in C5 is 40. Twenty plus 40 equals 60. Sixty is returned to the accumulator. The lack of the "#" symbol and a two-digit "number" indicate zero page mode.

All the commands I listed with stars work in any of these modes. The others work in other modes. None of the starless commands work in immediate mode, but some work in absolute or zero page modes.

Sometimes, especially in games, you will want to have a command access more than location, depending on a variable. For example, POKE 7680+X,42. This is quite common, very simple (and fun, I might add).

Absolute Indexing—This is the group of modes which allow you to add a variable to the address you are working with. There are several modes of absolute indexing, but some of them are above our heads at this point. For now, let's stick with absolute X and Y index modes. These modes are indicated by a ",X" or ",Y" after the address of the command.

First, let us think of the accumulator, the X and the Y indexes as the variables A, X, and Y respectively.

Example: ADC $1C00,X.
A=20. X=06. 1C06=15.
1C00+06(X)= 1C06.
15 (1C06)+20=35. Thirty-five is returned to A (the accumulator).

Example: ADC $CO,X.
A=20. X=05. C5=40.
C0+05(X)=C5.
40(C5)+20(A)=60. Sixty is returned to the accumulator.

NOTE: *In these modes, X or Y is added to the* address, *never the value. Only ",Y" is legal for some commands. See the PRG to see which ones it will work with.*

Just Absolute—Some commands run only in absolute or zero page mode. These are ASL, EOR, DEC, INC, JMP, JSR, LSR, ROL, ROR, STA, STX, and STY. This is very simple. Refer to the absolute and zero page sections above.

They Just Do—A few more commands stand alone (no "number" is needed.) These are BRK, CLC, CLD, CLI, CLV, DEX, DEY, INX, INY, NOP, PHA, PHP, PLA, PLP, RTI, RTS, SEC, SED, SEI, TAX, TAY, TSX, TXA, TXS, and TYA. They just do their job, no questions asked.

Relative—There are eight branching commands. They are very similar to IF . . . THEN . . . in BASIC. They are BCC, BCS, BEQ, BMI, BNE, BPL, BVC, and BVS. When using an assembler, these commands look like "absolute mode only" commands. But, *they are not!* I will explain how branching works later, but, briefly, instead of an address, the "number" is relative to the current address. "Relative" simply means that the jump is measured from the location containing this command, not just giving the address of the destination and jumping there (absolute mode). You can go 128 backwards and 127 forward. Without an assembler, you cannot break this rule. Using an assembler, if you break this rule the assembler should refuse to assemble it.

Flags

These commands branch to another part of the program when certain "flags" are set (1) or clear (0), depending on which command you are using. Flags are not hard to understand. The status register is one byte (eight bits). Each bit is a flag. A flag is one bit which contains a one or a zero if a certain thing does or does not happen, or is or is not set or clear.

Only four flags have machine language branching commands built around them. These are the carry, zero, negative, and overflow flags. If the result of an operation is zero, for example, the zero flag is set. This would affect the BEQ and BNE commands. If the result of an operation is negative, the negative flag is set (See BMI and BPL). If the result of an operation is higher than FF or lower than 00, the carry flag is set. The overflow flag is too complicated to understand at this point. You will learn to use the overflow flag as you become more proficient at machine language.

Two more bits, the decimal mode and interrupt disable bits, serve functions. If the interrupt disable bit is set, the VIC will cease to interrupt until the bit is cleared. If the decimal mode bit is set, the VIC goes into decimal mode.

A Few Explanations

Here are a few explanations about how certain difficult concepts work.

Shift-ASL,LSR-Shift one bit one direction. If the 128 bit is shifted left, or the 1 bit is shifted right, then the carry flag is set and that bit is forgotten. Example: LSR $1C07. 1C07=33. Return is 16, and the carry flag is set.
(00100001→00010000+ Carry)

Rotate-ROL,ROR-Same as above, except the bit is not forgotten and the carry flag supplies the incoming bit.
(┌→00100001→00010000+ Carry ┐)

NOTE: *If no "number" is given, accumulator is implied, on both of the above.*

Compare-CMP,CPX,CPY. The value given is compared to A,X,Y and the correct flag is set (zero, negative, etc.). To "compare" is to give the difference between two numbers, or, in this case, to set or clear the correct flags. Example: CMP #$20. A=40. 40−20=20. Zero flag is cleared, the negative flag is cleared, and the carry flag is cleared.

A Few Reminders

Unless working with an assembler, you will not be able to write the commands you just learned as words. Instead, you'll have to POKE the programs in place. But, even if you don't have an assembler, take heart. Keep reading, and in the end, I'll show you how to convert the word commands to POKE statements. Just read about how to make programs, and instead of typing them in, just write them on paper. When you get to the end, convert them to POKE statements and you're on your way. But, I do suggest that you buy the *VIC 20 Programmer's Reference Guide.*

Also, in the *VIC 20 Programmer's Reference Guide,* a memory map is on pages 170-177. A "memory map" gives all the locations in memory, and tells what they do.

How To Build Programs

Machine language programs are built logically. For example, to store a 0F in 1E00 (the first location in screen memory, 7680 in decimal), you first look over the commands. So you must do something like LDX #$0F, then STX $1E00. LDX loads the X register, STX stores the value of X somewhere.

Now, let's try to subtract 50 from 80. Look over the commands. Write a short program to do the subtraction. Do it on your assembler.

```
Here is mine—1C00 LDA #$80
             1C02 SBC #$50
             1C04 BRK
```

Run it from 1C00. The answer is in the accumulator now. Here is how it worked. LDA puts a number in A. SBC subtracts a number from A. BRK stops the program. Try doing the same program using the command ADC instead of SBC.

Experiment. Type in short programs like the one above, but use different commands and see what happens. It's really fun!

Next time, we will build more complex programs. Between now and then, practice. Experiment. Discover. Try new things. Don't worry, you can't hurt the computer from the keyboard (short of using a hammer to type)! C

John Karcher, age 15, is a home-schooled student who lives in Colmar, Pennsylvania.

Color Test for the Commodore 64

by Ross Cherednik

A little often goes a long way, as is the case with the program at hand. The twelve-line program displays on the screen all 4,096 possible combinations of border, background and print colors. With a running time just under two hours, each combination is on the screen about a second and a half. This gives enough time to quickly assess the clarity and aesthetic effects of a combination. Pressing RUN/STOP will freeze a particular combination on the screen. Typing CONT allows the program to continue. Surprising combinations turn out to work quite well with each other and can create exciting visual effects.

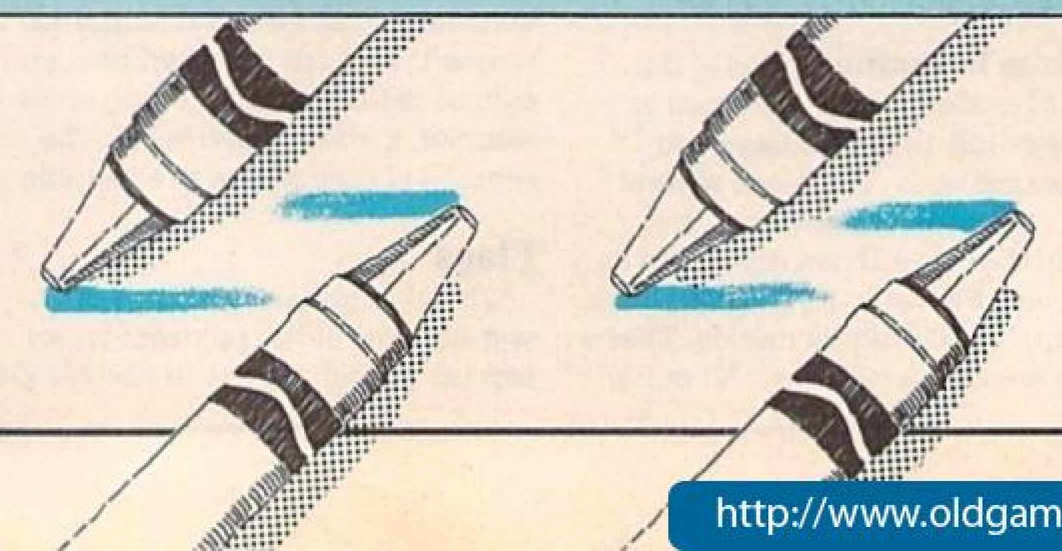
To run only border and background pairs and omit the print, simply delete lines 40, 70, 80, 90, and 100.

Explanation of Program

Line 10 clears the screen.
Lines 20-40 prepare the 16 colors for use.
Line 50 is the border color.
Line 60 is the background color.
Lines 70 and 80 spell "COLOR TEST".
Line 90 colors the print.
Line 100 changes print color.
Line 110 changes background color.
Line 120 changes border color.

```
5 REM ** PROGRAM BY ROSS CHEREDNIK **
10 PRINT"♥"
20 FOR A=0 TO 15
30 FOR B=0 TO 15
40 FOR C=0 TO 15
50 POKE 53280, A
60 POKE 53281, B
70 POKE 1394,3:POKE 1395,15:POKE 1396,12:
   POKE 1397,15:POKE 1398,18
80 POKE 1400,20:POKE 1401,5:POKE 1402,19:
   POKE 1403,20
90 FOR J=0 TO 9:POKE (55666+J),C:NEXT J
100 FOR X=1 TO 1000:NEXT X:NEXT C
110 FOR X=1 TO 1000:NEXT X:NEXT B
120 FOR X=1 TO 1000:NEXT X:NEXT A
```

C



64 Typer

by Judy Runge

Here's a short program that will let you use your Commodore 64 and printer as a typewriter! Word processors are great for typing letters and such, but this program will let you fill out forms and other things which are better suited to a typewriter than a word processor. It will print as you type, both on your printer and on your display screen, and will display a cursor and ruler on your screen to show you which column will print next. When you've finished typing, just press CTRL-c (no dash, just the control key and the lowercase "c" together) and you will return to BASIC.

For those of you who are just learning to program, I've included the following description of what each line does:

Line	Description
110	Initialize the line counter.
120	Open the printer channel.
130	Switch to lower case, disable the SHIFT/COMMODORE key, and clear the screen.
140	Print the ruler.
150	Here we print the cursor character. Then we back up one column so the next character will print over the cursor.
160	Get a character. If none are there, wait until one shows up!
170	See if the character is a CONTROL-c. If so, we want to quit.
180	If the character is a RETURN we have to add one to our line counter and then print a "blank" to erase our screen cursor character.
190	Now we print the character on both the screen, and to the printer.
200	If this is the end of the screen, then we reset our line counter and print a new ruler.
210	Go back and get another character.
220	This is where we come to quit. To make sure that we exit our program in an orderly fashion, we will switch back to uppercase and re-enable the SHIFT/COMMODORE key. Finally, we clear the screen before we end.

64 Typer

```
100 REM TYPER
110 LINES=0
120 OPEN 14,4,8
130 PRINT CHR$(14),CHR$(8),
    CHR$(147)
140 PRINT"....:....1....
    :....2....:....3....:....4"
150 PRINT CHR$(127);
    :POKE 211,(PEEK(211)-1)
160 GET A$:IF A$=""GOTO 160
170 IF A$=CHR$(3)GOTO 220
180 IF A$=CHR$(13)THEN
    LINES=LINES+1:PRINT" ";
190 PRINT A$;:PRINT#14,A$;
200 IF LINES=23 THEN:LINES=0
    :GOTO 140
210 GOTO 150
220 CLOSE 14:PRINT CHR$(9),
    CHR$(142),CHR$(147)
230 END
```

①

C

Tape Images on the Commodore 64

by
Elizabeth
Deal

Watch 64 Load

```
100 REM--------------------
110 REM C64 TAPE SNAPSHOTS
    ELIZABETH DEAL
120 REM--------------------
130 REM FILL C64 WITH $E5 (229),
     SAVE. REWIND TAPE. PUSH
    PLAY.
140 REM RUN THIS. USE 'CTRL' TO
    KILL THE MOTION (OR UNCRASH),
150 REM AND TO TAKE STOCK OF
    WHAT FACES YOU. HAVE FUN!
160 :
170 IF C=0 THEN DIM V(19)
    :FOR J=896 TO 941:READ V
    :C=C+V:POKE J,V:NEXT J
180 IF C<>6679 THEN STOP
190 X$="[SPACE39]"
200 X$=X$+CHR$(157)+CHR$(148)+" "
210 PRINT CHR$(147)CHR$(14);
220 FOR J=0 TO 19:PRINT X$:NEXT J
230 SYS 896: REM END
240 DATA 120,160,0,169,4,132,186,
    133
250 DATA 187,174,1,220,224,251,
    240,28
260 DATA 173,13,220,41,16,240,3,
    169
270 DATA 122,44,169,99,145,186,
    230,186
280 DATA 208,238,230,187,166,187,
    224,7
290 DATA 144,230,176,213,88,96
300 :
310 REM *** DON'T TYPE  BEYOND
    THIS POINT UNLESS YOU MUST
    ***
320 REM '23' IN 'IFV>=23' (LINE
    440) MAY NEED A FIX IF
    THINGS DON'T
330 REM WORK IN 1MINUTE. USE THE
    SMALLEST OF THE BIGGEST
    NUMBERS.
340 :
350 L$="[LEFT2,RVS]"
360 S$="[SPACE2]"
370 R$="[RVOFF,RIGHT]"
380 PRINT"10 SEC. THINKING"CHR$
    (19);
390 A=1024:AA=A:BY=A:MF=1:E=1:K=0
    :P=0:V$=""
400 FOR J=A TO A+767
410 IF PEEK(J)<>122 THEN PRINT
    CHR$(29);:NEXT J:GOTO 470
420 V=J-AA-1:PA=AA:AA=J
430 IF V>2 THEN PRINT
    L$RIGHT$(S$+MID$(STR$(V),2),
    2);
440 IF MF THEN IF V>=23 THEN
    BY=PA:MX=V:MF=0
450 IF E THEN IF MF=0 THEN V(K)=V
    :K=K+1:IF K=20 THEN E=0
460 PRINT R$;:NEXT J
470 IF E OR NOT V(0)>V(1)THEN
    PRINT CHR$(19)"CAN'T SEE -
    LEADER? GAP?";:GOTO 540
480 POKE BY,42:PRINT" * STARTS A
    BYTE[SPACE4]"
490 FOR J=2 TO 16 STEP 2:X$="0"
500 IF V(J)>V(J+1)THEN X$="1"
    :IF J<18 THEN P=P+1
510 V$=X$+V$:NEXT J:PRINT V$;
520 PRINT",PARITY";
    :PP=-(V(18)>V(19)):PRINT PP;
530 IF(1 AND P)=PP THEN
    PRINT"MUFFED.UP";
540 INPUT"[SPACE3]DO MORE";Y$
    :IF ASC(Y$)=89 GOTO 170
560 REM--------------------
```

Tape Images *Continued*

This program permits you to watch the cassette tape in action. It uses a crude timing device, without clocks, interrupts and turning off the screen. It shouldn't really work at all—we're working on a time-sloppiness scale of about 30:1. But every sample I've taken gave correct results. The Commodore recording method is foolproof enough that while I would never trust LOAD to this sort of a routine, it is just fine for learning what is going on, normally behind a blank screen.

Now you can see the show in real time (well, almost). Just follow the directions in the REM lines. I strongly recommend that you do not type in the second part of the listing (below DATA lines), but, rather, figure out for yourself what is happening. It's more fun that way. In case you fail to see the patterns, the BASIC program is there to unravel some mysteries.

As far as I can tell, nothing can go really wrong. If the machine seems crashed—and it will if you RUN before pressing the PLAY—the CONTROL (CTRL) key will get you out of trouble. It is also used to stop the show.

The final INPUT statement sometimes appears over other information on the screen. Don't worry about it, just type Y or some other letter and the 64 will take care of the rest.

If you are interested in the microsecond details of this thing, here are three good sources for further snooping:

one Raeto West, *Programming the Pet/CBM,* COMPUTE! Books.

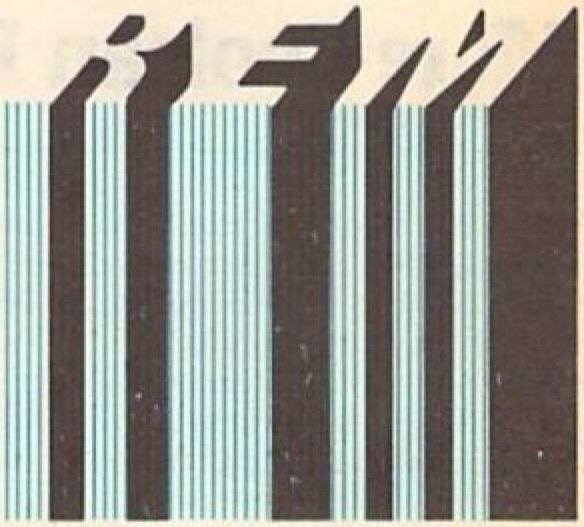
two Nick Hampshire, *The PET Revealed,* Computabits in the U.K.

three C64 ROMs.

C

Fun with REM

by John Stilwell



During my wanderings through the wonderful world of 2.0 BASIC, I've run across a lot of unique combinations. The discovery of these tricks is half the fun. Finding uses for them is the other half.

The REM statement has got to be the most neglected of all the commands. In college, students are taught "structured" programming stressing the importance of in-code documentation or REMs. Unfortunately, most home computer users have never had any classes and rarely have any documentation past the "written by" line. This is probably why the REM's secret uses have stayed a secret for so long.

Probably the flashiest thing you can do is make a line statement invisible. We accomplish this by putting DELETE characters in a REM statement. Setting this up is not very hard.

First of all, take a line, any line, and add a REM with two quotation marks in it.

```
10 PRINT"THIS IS A TEST":REM" "
```

Now put the cursor on the last quotation mark and push the insert key approximately twenty times. The computer now will print the character code of any of the commands! So push the DELETE key twenty times. If all has gone correctly, there should be twenty T's inside dark squares. If so, push RETURN to enter the line and we're done.

List line ten and you should see only "10 PRINT". Where's the rest of the line? It was there a moment ago. To prove line ten is okay, run it. If you want to erase the entire line, merely add more DELETEs.

Another trick with the REM makes for a simple list killer. All you have to do is add a shifted "L".

```
5 PRINT"ANOTHER TEST":REM L
```

During a listing, whenever one of these REMs are encountered, the listing will stop and you'll set a syntax error. Yet the program will run fine.

Go back to line five and change the shifted "L" to a shifted "M". List line five and the "M" has changed to a "FOR". This is because all of the BASIC commands are stored in memory as one-letter tokens. The token for "FOR" just happens to have the same ASCII code as the shifted "M".

If something is inside quotation marks, the computer looks at it one way. If a character is outside, it's considered to be a token. Our REM statement doesn't have quotation marks in it so the computer tries to untokenize our shifted "L". The reason we get the syntax error is because there is no command that matches this token. Thus the computer bombs when it sees the shifted "L".

If there is a program that you want to keep secret, scatter a few of the REMs throughout your code. It may not keep people out forever but it should slow them down. Especially if you make the line invisible first. C

More Boolean Foolin'

by Marilyn Sallee

Jim Butterfield's article, "Foolin' with Boolean", in the Summer, 1983, issue introduced you to the Boolean variables, but there is even more to the Boolean truth tables! Crunching your programs to make a whole series of IF... THEN statements into one statement is done by combining the Boolean truths with an ON statement.

The Boolean in your Commodore computer has the power to look at an arithmetic expression as a true or false statement. For example, in the statement X=(Y=7), if Y=7 is true then Boolean says X=−1 (true value). If Y is not 7, then X=0 (false value). Butterfield showed you all sorts of interesting things you can do with this handy trick—use it in IF... THEN statements and for decision arrays, for instance. Another powerful way to use it is in an ON statement.

Look at the following subroutine:

```
10 PRINT"ADD",."SUBTRACT",,
   "MULTIPLY",,"DIVIDE",,,,
   "CHOOSE ONE"
20 GET A$: IF A$="" THEN 20
30 IF A$="A" THEN 100
40 IF A$="S" THEN 200
50 IF A$="M" THEN 300
60 IF A$="D" THEN 400
70 GOTO 10
```

This is a common way to branch from a menu of choices. But the Boolean truth tables can combine lines 30,40,50,60, and 70 all into one. Here is the new line 30:

```
30 ON-(A$="A")-2*(A$="S")-3*(A$=
   "M")-4*(A$="D")GOTO100,200,
   300,400:GOTO10
```

Looks confusing. Let's break this down.

First, remember that if the arithmetic expression is true, it's given the value of minus one. If it's false, it is given the value of zero. But the ON statement only takes positive whole numbers. The first thing we have to do is make that truth value of minus one into a positive whole number. That's what the minus numbers between the parentheses do. (−1)*(−1)=+1, (−2)*(−1)=+2, (−3)*(−1)=+3, and so on.

The other important point is that false arithmetic expressions are given the value of zero, and anything multiplied by zero is zero. So (−1)*(0)=0, (−2)*(0)=0, (−3)*(0)=0, and so on.

Since the GET statement can only return one character, just one of the expressions inside the parentheses can be true at a time. If A$="A" then it cannot be "S", "M" or "D". So if one expression is true, then the ON statement can go to only one place, the line number selected by that true statement. All the other (false) values are zero.

One situation you have to watch out for is if all the expressions are false. In that case the ON statement becomes ON 0 THEN... what. That is what the GOTO 10 at the end is for.

But just because I used a GET statement does not mean that is the only choice. Let's look at an example:

```
10 INPUT"[CLEAR]WHAT MONTH";M$
15 M$=LEFT$(M$,3)
20 ON-(M$="JAN")-2*(M$="FEB")-3*
   (M$="MAR")-4*(M$="APR")GOTO 1
   00,200,300,400
30 ON-(M$="MAY")-2*(M$="JUN")-3*
   (M$="JUL")-4*(M$="AUG")GOTO 5
   00,600,700,800
40 ON-(M$="SEP")-2*(M$="OCT")-3*
   (M$="NOV")-4*(M$="DEC")GOTO 9
   00,1000,1100,1200
```

This subroutine allows you to check a whole string to find the one that is true, and then branch from there. If none is true it goes back to the original INPUT. Line 15 allows the user to input the whole month's name, like August, and the program will chop it off to Aug.

You can string as many of these ON statements together like this as suits your purpose. But remember that each line must send the ON statement to the proper GOTO for the true expression. This is why in line 30, even though May is the fifth month, it is the first expression being tested in that line. If it is true then you want the ON statement to GOTO the first line assignment, not the fifth.

This type of Boolean truth variable in an ON statement is most useful wherever you have several IF... THEN statements in a row.

Boolean

```
10 INPUT"[CLEAR]WHAT MONTH";M$
15 M$=LEFT$(M$,3)
20 ON-(M$="JAN")-2*(M$="FEB")-3*
   (M$="MAR")-4*(M$="APR")GOTO 1
   00,200,300,400
30 ON-(M$="MAY")-2*(M$="JUN")-3*
   (M$="JUL")-4*(M$="AUG")GOTO 5
   00,600,700,800
40 ON-(M$="SEP")-2*(M$="OCT")-3*
   (M$="NOV")-4*(M$="DEC")GOTO 9
   00,1000,1100,1200
100 PRINT"[DOWN2]JANUARY IS THE
    FIRST[SPACE2]MONTH AND HAS 3
    1 DAYS":GOTO 2000
200 PRINT"[DOWN2]FEBRUARY IS THE
     SECOND MONTH AND HAS 28 OR
    29 DAYS":GOTO 2000
300 PRINT"[DOWN2]MARCH IS THE TH
    IRD[SPACE2]MONTH AND HAS 31
    DAYS":GOTO 2000
400 PRINT"[DOWN2]APRIL IS THE FO
    URTH[SPACE2]MONTH AND HAS 30
     DAYS":COTO 2000
500 PRINT"[DOWN2]MAY IS THE FIFT
    H[SPACE2]MONTH AND HAS 31 DA
    YS":GOTO 2000
600 PRINT"[DOWN2]JUNE IS THE SIX
    TH[SPACE2]MONTH AND HAS 30 D
    AYS":GOTO 2000
700 PRINT"[DOWN2]JULY IS THE SEV
    ENTH[SPACE2]MONTH AND HAS 31
     DAYS":GOTO 2000
800 PRINT"[DOWN2]AUGUST IS THE E
    IGHTH[SPACE2]MONTH AND HAS 3
    1 DAYS":GOTO 2000
900 PRINT"[DOWN2]SEPTEMBER IS TH
    E NINTH[SPACE2]
    MONTH AND HAS 30 DAYS"
    :GOTO 2000
1000 PRINT"[DOWN2]OCTOBER IS THE
      TENTH[SPACE2]
     MONTH AND HAS 31 DAYS"
     :GOTO 2000
1100 PRINT"[DOWN2]NOVEMBER IS TH
     E ELEVENTH MONTH AND HAS 30
      DAYS":GOTO 2000
1200 PRINT"[DOWN2]DECEMBER IS TH
     E TWELFTH MONTH AND HAS 31
     DAYS":GOTO 2000
2000 FOR I=1 TO 3000:NEXT:GOTO 1(
```

c

Subscribe Now!

Get the inside story on Commodore computers, direct from Commodore. Discover all the latest developments in software, hardware and books… learn more about computing… get many programs to type and save. You get all this and more when you subscribe to Commodore's own user magazines, *Power/Play* and *Commodore Microcomputers.*

DON'T MISS ANOTHER ISSUE!
Use the subscription card enclosed to
SUBSCRIBE NOW!

Reading the Commodore 64 Keyboard

by Ray Sidney

Have you ever wondered exactly how to read the keyboard on the Commodore 64 without using the Kernal and BASIC subroutines that are built into the machine? Perhaps you want to be able to tell whether the user is holding down the <RETURN> key (not just whether <RETURN> has been pushed, but whether it is down at a given moment). The *Programmer's Reference Guide* has a bit to say, but it's not really very comprehensive on the subject. So here it is, folks, the way to directly read the keyboard from BASIC or from machine language.

First, you have to POKE a specific number into memory address 56320 ($DC00). Then you read memory address 56321 ($DC01) and mask the desired bits and check for zeros. Now, let's see exactly how to do all that. Look at Table 1. On the left, it has the number that you must POKE into 56320 for a certain key. Then read address 56321, and logically AND that value with the specified mask shown on the top row. Here's an example to check if the "C" key is being held down:

```
10 POKE 56320,251:IF (PEEK(56321)AND16)=0
THEN PRINT "C"
```

You can see that first it POKEs a 251 into 56320 (251 is the number on the left side corresponding to "C") and then it ANDs the byte from address 56321 with 16 (16 is the number from the top corresponding to "C"). The only tricky part is that if a key is being held down, instead of producing a one in that bit, it causes a zero, and if it is not held down, you read a one. The exact same approach works in machine language.

LFT SHF represents the left shift key or the shift lock key. RGT SHF is the right shift key. COM represents the key with the Commodore logo on it. CTR is the control key. LFT RGT means the left/right cursor key. UP/DWN is the up/down cursor key. RET is return. The rest of the keys should be pretty obvious.

Table 1.

	128	64	32	16	8	4	2	1
254	UP DWN	F5	F3	F1	F7	LFT RGT	RET	INS DEL
253	LFT SHF	E	S	Z	4	A	W	3
251	X	T	F	C	6	D	R	5
247	V	U	H	B	8	G	Y	7
239	N	O	K	M	0	J	I	9
223	< ,	@	[:	> .	−	L	P	+
191	? /	↑	=	RGT SHF	CLR HOM	] ;	*	£
127	RUN STP	Q	COM	SPC. BAR	2	CTR	←	1

User Group Listing

ALABAMA

Huntsville PET Users Club
9002 Berclair Road
Huntsville, AL 35802
Contact: Hal Carey
Meetings: every 2nd
Thursday

Riverchase Commodore Users
Group
617 Grove St.
Birmingham, AL 35209
(205) 988-1078
Ken Browning

Wiregrass Micro-Computer
Society
Commodore SIG
109 Key Bend Rd.
Enterprise, AL 36330
(205) 347-7564
Bill Brown

Tiger Byte: E. Alabama
CBM 64
Users Group
c/o The Computer Store, Inc.
Midway Plaza
Opelika, AL 36801
Jack Parsons
1st & 3rd Wed. of Month

The Birmingham Commodore
Computer Club
Birmingham, AL
(205) 923-9260
Harry Jones

Commodore Club of Mobile
3868-H Rue Maison
Mobile, AL 36608
(205) 343-1178
Tom Wyatt
3rd Thurs. of month

Shoals Commodore Users
Group (SCUG)
209 Lakeshore Dr.
Muscle Shoals, AL 35661
Geo. Taylor
2nd & 4th Tues. of month

1920-A Avenue C
Brookly
Mobile, AL 36615
(205) 661-1973
Howard Crider

ALASKA

COMPOOH-T
c/o Box 118
Old Harbor, AK 99643
(907) 286-2213

Alaska 84 Computer Club
c/o Line 49 Management
P.O. Box 6043
Anchorage, AK 99502

First City Users Group
P.O. Box 6692
Ketchikan, AK 99901
(907) 225-5695
James Llanos

ARIZONA

VIC Users Group
2612 E. Covina
Mesa, AZ 85203
Contact: Paul Muffuletto

Catalina Commodore
Computer Club
2012 Avenida Guillermo
Tucson, AZ 85710
(602) 296-6766
George Pope
1st Tues. 7:30 p.m.

Central Arizona PET People
842 W. Calle del Norte
Chandler, AZ 85224
(602) 899-3622
Roy Schahrer

ACUG
c/o Home Computer Service
2028 W. Camelback Rd.
Phoenix, AZ 85015
(602) 249-1186
Dan Deacon
First Wed. of month

West Mesa VIC
2351 S. Standage
Mesa, AZ 85202
Kenneth S. Epstein

Arizona VIC 20-64 Users Club
232 W. 9th Place North
Mesa, AZ 85201
Donald Kipp

Arizona VIC & 64 Users
904 W. Marlboro Circle
Chandler, AZ 85224
(602) 963-6149
Tom Monson

ARKANSAS

Commodore/PET Users Club
Conway Middle School
Davis Street
Conway, AR 72032
Contact: Geneva Bowlin

Booneville 64 Club
c/o A. R. Hederich
Elementary School
401 W. 5th St.
Booneville, AR 72927
Mary Taff

The Siloam Commodore
Computer Club
P.O. Box 88
Siloam Springs, AR 72761
(501) 524-5624
Ken Emanualson

Russellville Commodore
User Group
401 S. Arlington Dr.
Russellville, AR 72801
(501) 967-1868
Bob Brazeal

Arkansas River Valley
Commodore Users
401 S. Arlington Dr.
Russellville, AR 72801
(501) 967-1868
Bob Brazeal

CALIFORNIA

SCPUG Southern California
PET Users Group
c/o Data Equipment
Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: First Tuesday of
each month

California VIC Users Group
c/o Data Equipment
Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: Second Tues. of
each month

Valley Computer Club
1913 Booth Road
Ceres, CA 95307

PUG of Silicon Valley
22355 Rancho Ventura Road
Cupertino, CA 95014

Lincoln Computer Club
750 E. Yosemite
Manteca, CA 95336
John Fung, Advisor

PET on the Air
525 Crestlake Drive
San Francisco, CA 94132
Max J. Babin, Secretary

PALS (Pets Around)
Livermore Society
886 South K
Livermore, CA 94550
(415) 449-1084
Every third Wednesday
7:30 p.m.
Contact: J. Johnson

SPHINX
7615 Leviston Ave.
El Cerrito, CA 94530
(415) 527-9286
Bill MacCracken

San Diego PUG
c/o D. Costarakis
3562 Union Street
(714) 235-7626
7 a.m.-4 p.m.

Walnut Creek PET
Users Club
1815 Ygnacio Valley Road
Walnut Creek, CA 94596

Jurupa Wizards
8700 Galena St.
Riverside, CA 92509
781-1731
Walter J. Scott

The Commodore Connection
2301 Mission St.
Santa Cruz, CA 95060
(408) 425-8054
Bud Massey

San Fernando Valley
Commodore Users Group
21208 Nashville
Chatsworth, CA 91311
(213) 709-4736
Tom Lynch
2nd Wed. 7:30

VACUUM
277 E. 10th Ave.
Chico, CA 95926
(916) 891-8085
Mike Casella
2nd Monday of month

VIC 20 Users Group
2791 McBride Ln. # 121
Santa Rosa, CA
(707) 575-9836
Tyson Verse

South Bay Commodore
Users Group
1402 W. 218th St.
Torrance, CA 90501
Contact: Earl Evans

Slo VIC 20/64 Computer Club
1766 9th St.
Los Osos, CA

The Diamond Bar R.O.P.
Users Club
2644 Amelgado
Haciendo Hgts., CA 91745
(213) 333-2645
Don McIntosh

Commodore Interest
Association
c/o Computer Data
14660 La Paz Dr.
Victorville, CA 92392
Mark Finley

Fairfield VIC 20 Club
1336 McKinley St.
Fairfield, CA 94533
(707) 427-0143
Al Brewer
1st & 3rd Tues. at 7 p.m.

Computer Barn
Computer Club
319 Main St.
Suite # 2
Salinas, CA 93901
757-0788
S. Mark Vanderbilt

Humboldt Commodore Group
P.O. Box 570
Arcata, CA 95521
R. Turner

Napa Valley Commodore
Computer Club
c/o Liberty Computerware
2680 Jefferson St.
Napa, CA 94558
(707) 252-6281
Mick Winter
1st & 3rd Mon. of month

S.D. East County C-64
User Group
6353 Lake Apopka Place
San Diego, CA 92119
(619) 698-7814
Linda Schwartz

Commodore Users Group
4237 Pulmeria Ct.
Santa Maria, CA 93455
(805) 937-4174
Gilbert Vela

Bay Area Home
Computer Asso.
1332 Pine St.
Walnut Creek, CA 94598
(415) 932-5447
Cliff Downing

Amateurs and Artesians
Computing
P.O. Box 682
Cobb, CA 95426

Manteca VIC 20 Users
Organization
429 N. Main St.
Manteca, CA 95336
Gene Rong

Pomona Valley Commodore
Users Group
1401 W. 9th, # 77
Pomona, CA 91766
(714) 620-8889
Mark Joerger
1st & 3rd Wed. of month 7 p.m.

VIC TORII-The VIC 20
Users Group
PSC # 1, Box 23467
APO San Francisco, CA 96230
Wesley Clark

The Valley Computer Club
2006 Magnolia Blvd.
Burbank, CA 91506
1st Wed. 7 p.m.

The Commodore Tech. Users
of Orange Co.
P.O. Box 1497
Costa Mesa, CA 92626
(714) 731-5195
Roger Fisher

VIC 20 Software
Exchange Club
10530 Sky Circle
Grass Valley, CA 95945
Daniel Upton

C-64 West Orange County
Users Group
P.O. Box 1457
Huntington Beach, CA 92647
(714) 842-4484
Philip Putman
2nd & 4th Tues. of month

Antelope Valley Commodore
Users Group
POB 4436
Lancaster, CA 93539
(805) 942-2626
James Haner
1st Saturday

Diablo Valley Commodore
Users Group
762 Ruth Dr.
Pleasant Hill, CA 94523
(415) 671-0145
Ben Braver
2nd & 4th Thurs. 7:30 p.m.

Commodore Connection
11652 Valverde Ave.
Riverside, CA 92505
(714) 689-7447
Tony Alvarez

CA. Area Commodore
Terminal
Users Society
C.A.C.T.U.S.
P.O. Box 1277
Alta Loma, CA 91701
Darrell Hall

20/64
P.O. Box 18473
San Jose, CA 95158
(408) 978-0546
1st Sun. of month (6-9 p.m.)

8120 Sundance Dr.
Orangevale, CA 95662
(916) 969-2028
Robyn Graves

Software 64
35 California Dr.
Burlingame, CA 94010
(415) 340-7115
Mario Abad

Sacramento Commodore
Users Group
8120 Sundance Dr.
Orangevale, CA 95662
(916) 969-2028
Robyn Graves

Peninsula Commodore
Users Group
549 Old County Rd.
San Carlos, CA 94070
(415) 593-7697
Timothy Very
2nd Thurs. of Month

San Francisco Commodore
Users Group
278-27th Ave. # 103
San Francisco, CA 94121
(415) 387-0225
Roger Tierce

South Bay Commodore 64
Users Group
P.O. Box 3193
San Ysidro, CA 95073

Commodore 64 West
Computer Club
2917 Colorado Ave.
Santa Monica, CA 90404
(213) 828-9308
Don Campbell

Sixty Fourum
4413 E. Iowa
Fresno, CA 93702
(209) 252-0392
Deb Christensen

C-64/VIC 20 Users Group
Pasadena City College
Cicadian Room
Pasadena, CA
(714) 593-4880
Chuck Cypher
7 p.m. 1st & 3rd Thursdays

Marin Commodore
Computer Club
620 Del Ganado Rd.
San Rafael, CA
(415) 479-0426
2nd Wed. of month 7:30 p.m.

COLORADO

VICKIMPET Users Group
4 Waring Lane, Greenwood
Village
Littleton, CO 80121
Contact: Louis Roehrs

Colorado Commodore
Computer Club
2187 S. Golden Ct.
Denver, CO 80227
986-0577
Jack Moss
Meet: 2nd Wed.

Commodore Users Group
Box 377
Aspen, CO 81612
(303) 925-5604
Ray Brooks
1st Monday in the evenings

CONNECTICUT

John F. Garbarino
Skiff Lane Masons Island
Mystic, CT 06355
(203) 536-9789

Commodore User Club
Wethersfield High School
411 Wolcott Hill Road
Wethersfield, CT 06109
Contact: Daniel G. Spaneas

VIC Users Club
c/o Edward Barszczewski
22 Tunxis Road
West Hartford, CT 06107

New London County
Commodore Club
Doolittle Road
Preston, CT 06360
Contact: Dr. Walter Doolittle

The Commodore East Users Group
165 B S. Bigelow Rd.
Hampton, CT 06247
(203) 455-0108

DISTRICT OF COLUMBIA

USO Computer Club
USO Outreach Center
207 Beyer Rd., SW
Washington, DC 20332
Steven Guenther

FLORIDA

Jacksonville Area
PET Society
401 Monument Road, #177
Jacksonville, FL 32211

Richard Prestien
6278 SW 14th Street
Miami, FL 33144

South Florida
PET Users Group
Dave Young
7170 S.W. 11th
West Hollywood, FL 33023
(305) 987-6982

PETs and Friends
129 NE 44 St.
Miami, FL 33137
Richard Plumer

Sun Coast VICs
P.O. Box 1042
Indian Rocks Beach, FL
33535
Mark Weddell

Bay Commodore Users Group
c/o Gulf Coast Computer
Exchange
241 N. Tyndall Pkwy.
P.O. Box 6215
Panama City, FL 32401
(904) 785-6441
Richard Scofield

Gainesville Commodore
Userp Club
3604-20A SW 31st Dr.
Gainesville, FL 32608
Louis Wallace

Brandon Users Group
108 Anglewood Dr.
Brandon, FL 33511
(813) 685-5138
Paul Daugherty

Brandon Commodore Users Group
414 E. Lumsden Rd.
Brandon, FL 33511

Gainesville Commodore
Users Group
Santa Fe Community College
Gainesville, FL 32602
James E. Birdsell

Commodore Computer Club
P.O. Box 21138
St. Petersburg, FL 33742
(813) 522-2547
Chuck Fechko

Commodore Users Group
545 E. Park Ave.
Apt. #2
Tallahassee, FL 32301
(904) 224-6286
Jim Neill

The Commodore Connection
P.O. Box 6684
West Palm Beach, FL 33405

El Shift OH
P.O. Box 548
Cocoa, FL 32922
Mike Schnoke
Sat. mornings/every 4 to 6 weeks

Miami 20/64
12911 S.W. 49th St.
Miami, FL 33175
(305) 226-1185

Tampa Bay Commodore
Computer Club
10208 N. 30th St.
Tampa, FL 33612
(813) 977-0877

Commodore Computer Club
P.O. Box 9726
Jacksonville, FL 32208
(904) 764-5457
David Phillips
2nd & 4th Tues. of Month

VIC/64 Heartland Users Group
1220 Bartow Rd. #23
Lakeland, FL 33801
(813) 666-2132
Tom Keough
4th Wed. of Month at PRC

64 Educators Users Group South
FDLRS-South
9220 S.W. 52nd Terrace
Miami, FL 33165
(305) 274-3501
Dr. Eydie Sloane

64 Educators Users Group North
16330 N.E. 2nd Ave.
North Miami Beach, FL 33162
(305) 944-5548
Robert Figueroa

Suncoast 64S
c/o Little Professor Book Center
2395 U.S. 19 North
Palm Harbor, FL 33563
(813) 785-1036
Curtis Miller

Lakeland VIC 20 Users Group
2450 Shady Acres Dr.
Mulberry, FL 33860

Broward Commodore Users Group
13 Spinning Wheel Lane
Tamarac, FL 33319
(305) 726-4390
Lewis Horn

GEORGIA

VIC Educators Users Group
Cherokee County Schools
110 Academy St.
Canton, GA 30114
Dr. Al Evans

Bldg. 68, FLETC
Glynco, GA 31524
Richard L. Young

VIC-tims
P.O. Box 467052
Atlanta, GA 30346
(404) 922-7088
Eric Ellison

Golden Isles Commodore Users Club
Bldg. 68, FLETC
Glynco, GA 31524
Richard L. Young

Commodore Club of Augusta
1011 River Ridge Rd.
Apt. #14-A
Augusta, GA 30909
David Dumas

Dataswapper Users Group
1794 Alabama Ave.
Albany, GA 31705
(912) 436-5596
David Via

HAWAII

Commodore Users Group of Honolulu
c/o PSH
824 Bannister St.
Honolulu, HI
(808) 848-2088
3rd Fri. every month

20/64 Hawaii
P.O. Box 966
Kailua, HI 96734
Wes Goodpaster

Commodore Users Group of Honolulu
1626 Wilder #701
Honolulu, HI 96822
(808) 848-2088
Jay Calvin (808) 944-9380

IDAHO

GHS Computer Club
c/o Grangeville High School
910 S. D St.
Grangeville, ID 83530
Don Kissinger

S.R.H.S. Computer Club
c/o Salmon River H.S.
Riggins, ID 83549
Barney Foster

Commodore Users
548 E. Center
Pocatello, ID 83201
(208) 233-0670
Leroy Jones

Eagle Rock Commodore Users Group
900 S. Emerson
Idaho Falls, ID 83401
Nancy J. Picker

64 Bug (Boise Users Group)
P.O. Box 276
Boise, ID 83701
(208) 344-6302
John Rosecrans

ILLINOIS

Shelly Wernikoff
2731 N. Milwaukee
Avenue
Chicago, IL 60647

VIC 20/64 Users Support
Group
c/o David R. Tarvin
114 S. Clark Street
Pana, IL 62557
(217) 562-4568

Central Illinois PET User
Group
635 Maple
Mt. Zion, IL 62549
(217) 864-5320
Contact: Jim Oldfield

ASM/TED User Group
200 S. Century
Rantoul, IL 61866
(217) 893-4577
Contact: Brant Anderson

PET VIC Club (PVC)
40 S. Lincoln
Mundelein, IL 60060
Contact: Paul Schmidt,
President

Rockford Area PET Users
Group
1608 Benton Street
Rockford, IL 61107

Commodore Users Club
1707 East Main St.
Olney, IL 62450
Contact: David E. Lawless

VIC Chicago Club
3822 N. Bell Ave.
Chicago, IL 60618
John L. Rosengarten

Chicago Commodore 64
Users & Exchange Group
P.O. Box 14233
Chicago, IL 60614
Jim Robinson

Fox Valley PET Users
Group
833 Willow St.
Lake in the Hills, IL 60102
(312) 658-7321
Art DeKneef

The Commodore 64 Users
Group
P.O. Box 572
Glen Ellyn, IL 60137
(312) 790-4320
Gus Pagnotta

RAP 64/VIC Regional
Assoc. of Programmers
10721 S. Lamon
Oak Lawn, IL 60453
Bob Hughes

The Kankakee Hackers
RR #1, Box 279
St. Anne, IL 60964
(815) 933-4407
Rich Westerman

WIPUG
Rt. 5, Box 75
Quincy, IL 62301
(217) 656-3671
Edward Mills

Papug-Peoria Area Pet Users Group
6 Apple Tree Lane
East Peoria, IL 61611
(309) 673-6635
Max Taylor
2nd Fri. of Month

McHenry County Commodore Club
4900 S. Route 31
Crystal Lake, IL 60014
(815) 455-3942
John Katkus
2nd Sat. of month 9 to 12 a.m.

INDIANA

PET/64 Users
10136 E. 96th St.
Indianapolis, IN 46256
(317) 842-6353
Jerry Brinson

Cardinal Sales
6225 Coffman Road
Indianapolis, IN 46268
(317) 298-9650
Contact: Carol Wheeler

CHUG (Commodore
Hardware Users Group)
12104 Meadow Lane
Oaklandon, IN 46236
Contact: Ted Powell

VIC Indy Club
P.O. Box 11543
Indianapolis, IN 46201
(317) 898-8023
Ken Ralston

Northern Indiana
Commodore Enthusiasts
927 S. 26th St.
South Bend, IN 46615
Eric R. Bean

Commodore Users Group
1020 Michigan Ave.
Logansport, IN 46947
(219) 722-5205
Mark Bender

Computer Workshop VIC 20/64 Club
282 S. 600 W.
Hebron, IN 46341
(219) 988-4535
Mary O'Bringer

The National Science Clubs
of America
Commodore Users Division
7704 Taft St.
Merrillville, IN 46410
Brian Lepley or Tom Vlasic

East Central Indiana VIC User Group
Rural Route #2
Portland, IN 47371
Stephen Erwin

National VIC 20 Program Exchange
102 Hickory Court
Portland, IN 47371
(219) 726-4202
Stephen Erwin

Commodore Computer Club
3814 Terra Trace
Evansville, IN 47711
(812) 477-0739
John Patrick, President

Commodore 64 Users Group
912 South Brown Ave.
Terre Haute, IN 47803
(812) 234-5099
Dennis Graham

Seymour Peekers
c/o D&L Camera Shop
108 N. Chestnut
Seymour, IN 47274
Dennis Peters

IOWA

Commodore User Group
114 8th St.
Ames, IA 50010

Quad City Commodore Club
1721 Grant St.
Bettendorf, IA 52722
(319) 355-2641
John Yigas

Siouxland Commodore Club
2700 Sheridan St.
Sioux City, IA 51104
(712) 258-7903
Gary Johnson
1st & 3rd Monday of month

421 W. 6th St.
Waterloo, IA 50702
(319) 232-1062
Frederick Volker

Commodore Computer Users
Group of Iowa
Box 3140
Des Moines, IA 50316
(515) 263-0963 or (515) 287-1378
Laura Miller

Commo-Hawk Commodore
Users Group
P.O. Box 2724
Cedar Rapids, IA 52406
Vern Rotert

KANSAS

Wichita Area PET Users Group
2231 Bullinger
Wichita, KS 67204
(316) 838-0518
Contact: Mel Zandler

Kansas Commodore
Computer Club
101 S. Burch
Olathe, KS 66061
Contact: Paul B. Howard

Commodore Users Group
6050 S. 183 St. West
Viola, KS 67149
Walter Lounsbery

Walnut Valley Commodore
User Group
1003 S. 2nd St.
Arkansas City, KS 67005
Bob Morris

KENTUCKY

VIC Connection
1010 S. Elm
Henderson, KY 42420
Jim Kemp

Louisville Users of Commodore KY
(LUCKY)
P.O. Box 22244
Louisville, KY 40222
(502) 425-2847
2nd Tues. of Month

The Bowling Green Commodore
Users Group
Route 11, Creekside Apt. #6
Bowling Green, KY 42101
(502) 781-9098
Alex Fitzpatrick

LOUISIANA

Franklin Parish Computer Club
#3 Fair Ave.
Winnisboro, LA 71295
James D. Mays, Sr.

NOVA
917 Gordon St.
New Orleans, LA 70117
(504) 948-7643
Kenneth McGruder, Sr.

VIC 20 Users Group
5064 Bowdon St.
Marrero, LA 70072
(504) 341-5305
Wayne D. Lowery, R.N.

64-Club News
5551 Corporate Blvd.
Suite 3L
Baton Rouge, LA 70808
(504) 766-7408
Tom Parsons
3rd Tues. of month at CWA

Commodore Users Group of Oachita
P.O. Box 175
Swaric, LA 71281
(318) 343-8044
Beckie Walker

Ark-La-Tex Commodore 64 Club
198 India Dr.
Shreveport, LA 71115
(318) 797-9702
Pete Whaley

Commodore 64 Users Group
P.O. Box 1422
Baton Rouge, LA 70821
Richard Hood
3rd Tues. of month

MAINE

COM-VICS (Commodore/VIC
Users Group)
RFD #1, Box 2086
Hebron, ME 04238
(207) 966-3641
Paul Lodge
1st Wed. & 3rd Thurs.

Your Commodore Users Group
Box 611
Westbrook, ME 04092
(207) 854-4579
Mike Procise

MARYLAND

Assoc. of Personal
Computer Users
5014 Rodman Road
Bethesda, MD 20016

Blue TUSK
700 East Joppa Road
Baltimore, MD 21204
Contact: Jim Hauff

House of Commodore
8835 Satyr Hill Road
Baltimore, MD 21234
Contact: Ernest J. Fischer

Long Lines Computer Club
323 N. Charles St., Rm. 201
Baltimore, MD 21201
Gene Moff

VIC & 64 Users Group
The Boyds Connection
21000 Clarksburg Rd.
Boyds, MD 20841
(301) 428-3174
Tom DeReggi

Rockville VIC/64 Users Group
P.O. Box 8805
Rockville, MD 20856
(301) 231-7823
Tom Pounds

The Compucats' Commodore
Computer Club
680 W. Bel Air Ave.
Aberdeen, MD 21001
(301) 272-0472
Betty Jane Schueler

Westinghouse BWI
Commodore User Group
Attn: L. Barron
Mail Stop 5156
P.O. Box 1693
Baltimore, MD 21203

HUG (Hagerstown Users Group)
23 Coventry Lane
Hagerstown, MD 21740
(301) 797-9728
Joseph Rutkowski

The Montgomery Ct. Commodore
Computer Soc.
P.O. Box 6444
Silver Springs, MD 20906
(301) 946-1564
Meryle Pounds

Commodore Users Group
of Annapolis
P.O. Box 9726
Arnold, MD 21012
(301) 974-4548
The Software Co.

MASSACHUSETTS

Eastern Massachusetts
VIC Users Group
c/o Frank Ordway
7 Flagg Road
Marlboro, MA 02173

VIC Users Group
c/o Ilene Hoffman-Sholar
193 Garden St.
Needham, MA 02192

Commodore Users Club
Stoughton High School
Stoughton, MA 02072
Contact: Mike Lennon

Berkshire PET Lovers
CBM Users Group
Taconic High
Pittsfield, MA 01201

The Boston Computer
Society
Three Center Plaza
Boston, MA 02108
(617) 367-8080
Mary E. McCann

Masspet Commodore Users Group
P.O. Box 307
East Taunton, MA 02718
David Rogers

Raytheon Commodore Users Group
Raytheon Company
Hartwell Rd. GRA-6
Bedford, MA 01730
John Rudy

Commodore 64 Users
Group of The Berkshires
184 Highland Ave.
Pittsfield, MA 01201
Ed Rucinski

VIC Interface Club
48 Van Cliff Ave.
Brockton, MA 02401
Bernie Robichaud

Cape Cod 64 Users Group
358 Forrest Rd.
S. Yarmouth, MA 02664
1 (800) 225-7136
Jim Close
(In MA. call) 1 (800) 352-7787

The Cursor Club
442 Mulpuf Rd.
Lunenburg, MA 01462
(617) 582-0529
John

Pioneer Valley VIC/64 Club
34 Bates St.
Westfield, MA 01085
(413) 562-1027
Marvin Yale
3rd Thurs. of month

MICHIGAN

David Liem
14361 Warwick Street
Detroit, MI 48223

VIC Users Club
University of Michigan
School of Public Health
Ann Arbor, MI 48109
Contact: John Gannon

Commodore User Club
32303 Columbus Drive
Warren, MI 48093
Contact: Robert Steinbrecher

Commodore Users Group
c/o Family Computer
3947 W. 12 Mile Rd.
Berkley, MI 48072

VIC for Business
6027 Orchard Ct.
Lansing, MI 48910
Mike Marotta

South Computer Club
South Jr. High School
45201 Owen
Belleville, MI 48111
Ronald Ruppert

Commodore Users Group
c/o Eaton Rapids Medical Clinic
101 Spicerville Hwy.
Eaton Rapids, MI 48827
Albert Meinke III, M.D.

South East Michigan Pet
Users Group
Box 214
Farmington, MI 48024
Norm Eisenberg

Commodore Computer Club
4106 Eastman Rd.
Midland, MI 48640
(517) 835-5130
John Walley
9:30 p.m. Sept/May

VIC, 64, PET Users Group
8439 Arlis Rd.
Union Lake, MI 48085
363-8539
Bert Searing

COMP
486 Michigan Ave.
Marysville, MI 48040
(313) 364-6804
M. Gauthier

Ann Arbor Commodore Users Group
Ann Arbor, MI 48103
(313) 994-4751
Art Shaw
3rd Tues. 7:30-10:00

DAB Computer Club
P.O. Box 542
Watervliet, MI 49098
(616) 463-5457
Dennis Burlingham

West Michigan Commodores
c/o R. Taber
1952 Cleveland Ave., S.W.
Wyoming, MI 49509
(616) 458-9724
Gene Traas

Debug
P.O. Box 196
Berrien Springs, MI 49103
(616) 471-1882
Herbert Edward
Last Thursday of Month

Jackson Commodore Computer Club
201 S. Grinnell St.
Jackson, MI 49203
Alfred Bruey
Last Thur. of Month 7:30 p.m.

SMCUG
1002 Pfau St.
Mankato, MI 56001
(507) 625-6942
Dean Otto

SEM 64
25015 Five Mile #3
Redford, MI 48239
(313) 537-4163
Gary Groeller

C.A.T.O.
17606 Valade
Riverview, MI 48192
Dean Tidwell

MINNESOTA

MUPET (Minnesota Users
of PET)
P.O. Box 179
Annandale, MN 55302
c/o Jon T. Minerich

Twin Cities Commodore
Computer Club
6623 Ives Lane
Maple Grove, MN 55369
(612) 424-2425
Contact: Rollie Schmidt

Brainerd Area Commodore Users Group
1219 S.E. 11th St.
Brainerd, MN 56401
(218) 829-0805
Norm Saavedra
1st Thurs. 6 p.m. & 3rd Sat. 10 a.m.

MISSISSIPPI

Commodore Biloxi
User Group (ComBUG)
Universal Computer Services
3002 Hwy. 90 East
Ocean Springs, MS 39564
(601) 875-1173
John Lassen

MISSOURI

KCPUG
(Commodore User Group of
Kansas City)

P.O. Box 36492
Kansas City, MO 64111
(816) 252-7628
Salvadore

Commodore User Group of St. Louis
Box 6653
St. Louis, MO 63125-0653
Dan Weidman, New Members
1541 Swallowtail Dr.
St. Louis, MO

VIC INFONET
P.O. Box 1069
Branson, MO 65616
(417) 334-6099
Jory Sherman

Worth County PET Users Group
Grant City, MO
(816) 564-3551
David Hardy

Mid-Missouri Commodore Club
1804 Vandiver Dr.
Columbia, MO 65201
(314) 474-4511
Phil Bishop

Joplin Commodore Computers Users Group
422 S. Florida Ave.
Joplin, MO 64801
R. D. Connely

MONTANA

Powder River Computer Club
Powder River County High School
Broadus, MT 59317
Contact: Jim Sampson

Commodore User Club
1109 West Broadway
Butte, MT 59701
Contact: Mike McCarthy

NEBRASKA

Greater Omaha Commodore 64 Users Group
2932 Leawood Dr.
Omaha, NE 68123
(402) 292-2753
Bob Quisenberry

NEVADA

Las Vegas PET Users
Suite 5-315
5130 E. Charleston Blvd.
Las Vegas, NV 89122
Gerald Hasty

Compu Club 64
4220 S. Maryland Parkway
Bldg. B—Suite 403
Las Vegas, NV 89109
(702) 369-7354
Cindy Springfield

NEW JERSEY

Commodore Friendly User Group
49 Hershey Rd.
Wayne, NJ 07470
(201) 696-8043
Rich Pinto/Colin Campbell

Somerset Users Club
49 Marcy Street
Somerset, NJ 08873
Contact: Robert Holzer

Educators Advisory
P.O. Box 186
Medford, NJ 08055
(609) 953-1200
John Handfield

VIC-TIMES
46 Wayne Street
Edison, NJ 08817
Thomas R. Molnar

VIC 20 User Group
67 Distler Ave.
W. Caldwell, NJ 07006
(201) 284-2281
G. M. Amin

VIC Software Development Club
77 Fomalhaut Ave.
Sewell, NJ 08080
H. P. Rosenberg

ACGNJ PET/VIC/CBM User Group
30 Riverview Terr.
Belle Mead, NJ 08502
(201) 359-3862
J. M. Pylka

South Jersey Commodore Users Club
46-B Monroe Path
Maple Shade, NJ 08052
(609) 667-9758
Mark Orthner

Parsippany Computer Group
51 Ferncliff Rd.
Morris Plains, NJ 07950
(201) 267-5231
Bob Searing

Information 64
16 W. Ridgewood Ave.
Ridgewood, NJ 07450
(201) 447-4432
Dave Garaffa

NEW HAMPSHIRE

Northern New England Computer Society
P.O. Box 69
Berlin, NH 03570

TBH VIC-NICs
P.O. Box 981
Salem, NH 03079

C-64 U.S.E.R.S. User Software Exchange Pro
P.O. Box 4022
Rochester, NH 03867
Paul Kyle

NEW MEXICO

Commodore Users Group
6212 Karlson, NE
Albuquerque, NM 87113
(505) 821-5812
Danny Byrne

NEW YORK

Capital District 64/VIC 20 Users Group
363 Hamilton St.
Albany, NY 12210
(518) 436-1190
Bill Pizer

Long Island PET Society
Ralph Bressler
Harborfields HS
Taylor Avenue
Greenlawn, NY 11740

PET User Club of Westchester
P.O. Box 1280
White Plains, NY 10602
Contact: Ben Meyer

LIVE (Long Island VIC Enthusiasts)
17 Picadilly Road
Great Neck, NY 11023
Contact: Arnold Friedman

Commodore Masters
25 Croton Ave.
Staten Island, NY 10301
Contact: Stephen Farkouh

VIC Users Club
76 Radford St.
Staten Island, NY 10314
Contact: Michael Frantz

West Chester County VIC Users Group
P.O. Box 146
Pelham, NY 10552
Joe Brown

SPUG
4782 Boston Post Rd.
Pelham, NY 10803
Paul Skipski

VIC 20 User Club
151-28 22nd Ave.
Whitestone, NY 11357
Jean F. Coppola

VIC 20 User Club
339 Park Ave.
Babylon, NY 11702
(516) 669-9126
Gary Overman

VIC User Group
1250 Ocean Ave.
Brooklyn, NY 11230
(212) 859-3030
Dr. Levitt

L&M Computer Club
VIC 20 & 64
4 Clinton St.
Tully, NY 13159
(315) 696-8904
Dick Mickelson

Commodore Users Group
1 Corwin Pl.
Lake Katrine, NY 12449
J. Richard Wright

VIC 20/Commodore 64 Users Group
31 Maple Dr.
Lindenhurst, NY 11757
(516) 957-1512
Pete Lobol

VIC Information Exchange Club
336 W. 23 St.
Deer Park, NY 11729
Tom Schlegel
SASE & phone please

New York Commodore Users Group
380 Riverside Dr., 7Q
New York, NY 10025
(212) 566-6250
Ben Tunkelang

Hudson Valley Commodore Club
1 Manor Dr.
Woodstock, NY 12498
F.S. Goh
1st Wednesday of month

LIVICS (Long Island VIC Society)
20 Spyglass Lane
East Setauket, NY 11733
(516) 751-7844
Lawrence Stefani

VIC Users Group
c/o Stoney Brook Learning Center
1424 Stoney Brook Rd.
Stoney Brook, NY 11790
(516) 751-1719
Robert Wurtzel

Poughkeepsie VIC User Group
2 Brooklands Farm Rd.
Poughkeepsie, NY 12601
(914) 462-4518
Joe Steinman

VIC 20 User Group
Paper Service Division
Kodak Park
Rochester, NY 14617
David Upham, Sr.

Manhattan 64
c/o Steve Lazarowitz
1440 Freeport Loop
Brooklyn, NY 11239
(212) 647-4266
Larry Thompson

Adirondack Commodore 64 Users Group
205 Woodlawn Ave.
Saratoga Springs, NY
(518) 584-8960
Paul Klompas

Rockland County Commodore Users Group
P.O. Box 573
Nanuet, NY 10965
Ross Garber

New York 64 Users Group
222 Thompson St.
New York, NY 10012
(212) 673-7241
Bruce Cohen

Finger Lakes Commodore Users Group
c/o Rose City Computer Associates
229 West Union St.
Newark, NY 14513
(315) 331-1185

The Commodore Users Group Rochester
78 Hardison Rd.
Rochester, NY 14617
(716) 544-5251
Tom Werenski
Phone Evenings between 7-10

Commodore Computer Club
Publications Dept.,
Grumman Aerospace
1111 Stewart Ave.
Bethpage, NY 11714
(516) 575-9558
Neil Threulsen

Hello, Central!
76-12 35th Ave.
Jackson Heights, NY 11372
Jared Sherman

VIC 20/64 Users Group
NYU Waverly Place
New York, NY 10003
(212) 358-5155
Lawrence Schulman

SCUG (Schenectady Commodore Users Group)
c/o the Video Connection
Canal Square
Schenectady, NY 12305
Timothy Davis
1st Mon. of Month

Commodore 64 Users Group
S.U.N.Y. at Oswego
Dept. of Industrial Arts
Oswego, NY 13126
John R. Boronkay

NORTH CAROLINA

Amateur Radio PET Users Group
P.O. Box 30694
Raleigh, NC 27622
Contact: Hank Roth

VIC Users Club
c/o David C. Fonenberry
Route 3, Box 351
Lincolnton, NC 28092

Microcomputer Users Club
Box 17142 Bethabara Sta.
Winston-Salem, NC 27116
Joel D. Brown

VIC Users Club
Rt. 11, Box 686
Hickory, NC 28601
Tim Gromlovits

Raleigh VIC 20/64 Users Group
410-D Delta Court
Cary, NC 27511
(919) 469-3862
Larry Diener

Down East Commodores
302 Belltown Rd.
Havelock, NC 28532
(919) 447-4536
Bruce Thedin

Tryon Commodore 64 Club
P.O. Box 1016
Tryon, NC 28782
(704) 859-6340
Robin Michael
1st Mon. of month at 7 p.m.

OHIO

Dayton Area PET User Group
933 Livingston Drive
Xenia, OH 45385
B. Worby, President
(513) 848-2065
J. Watson, Secretary
(513) 372-2052

Central Ohio PET Users Group
107 S. Westmoor Avenue
Columbus, OH 43204
(614) 274-0304
Contact: Philip H. Lynch

Commodore Computer Club of Toledo
734 Donna Drive
Temperance, MI 48182
Gerald Carter

Chillicothe Commodore Users Group
P.O. Box 211
Chillicothe, OH 45601
William A. Chaney

Licking County 64 Users Group
323 Schuler St.
Newark, OH 43055
(614) 345-1327

11433 Pearl Rd.
Strongsville, OH 44136
Paul M. Warner

C.P.U. Connection
P.O. Box 42032
Brook Park, OH 44142
Danni Hudak

SE Cleveland Commodore Crazies
18813 Harlan Dr.
Maple Heights, OH 44137
(216) 581-3099
Carl Skala

Commodore Users of
Blue Chip (Cincinnati)
816 Beecher St.
Cincinnati, OH 45206
(513) 961-6582
Ted Stalets

Southwestern Ohio Commodore
Users Group
P.O. Box 399117
Cincinnati, OH 45239
2nd Wed. of month at 7 p.m.

S.W.O.C.U.G. (SW. Ohio
Commodore Users Gp.)
8401 Wicklow Ave.
Cincinnati, OH 45236
Joe Beresford

OKLAHOMA

Southwest Oklahoma
Computer Club
c/o Commodore Chapter
P.O. Box 6646
Lawton, OK 73504
1:30 at Lawton City Library

Tulsa Area Commodore Users Group
Tulsa Computer Society
P.O. Box 15238
Tulsa, OK 74112
Annette Hinshaw

Commodore Oklahoma Users Club
4000 NW 14th St.
Oklahoma City, OK 73107
(405) 943-1370
Stanley B. Dow

Commodore Users
Box 268
Oklahoma City, OK 73101
Monte Maker, President

Commodore Users of Norman
209 Brookwood
Noble, OK 73068
Matt Hager

Commodore Users Group
Muskogee Computer Society
202 S. 12th St.
Muskogee, OK 74401
Steve Ford

OREGON

NW PET Users Group
John F. Jones
2134 N.E. 45th Avenue
Portland, OR 97213

U.S. Commodore Users Group
P.O. Box 2310
Roseburg, OR 97470
(503) 672-7591
Richard Tsukiji

Southern Oregon VIC/64
Users Group
3600 Madrona Lane
Medford, OR 97501
(503) 779-7631
James Powell

PENNSYLVANIA

PET User Group
Gene Beals
P.O. Box 371
Montgomeryville, PA 18936

Penn Conference Computer Club
c/o Penn Conference of SDA
720 Museum Road
Reading, PA 19611
Contact: Dan R. Knepp

PACS Commodore Users Group
LaSalle College
20th & Olney Ave.
Philadelphia, PA 19141
(215) 951-1258
Stephen Longo

Glen Schwartz
807 Avon
Philadelphia, PA 19116

Gene Planchak
4820 Anne Lane
Sharpsville, PA 15150
(412) 962-9682

PPG (Pittsburgh PET Group)
c/o Joel A. Casar, DMD
2015 Garrick Drive
Pittsburgh, PA 15235
(412) 371-2882

Westmoreland Commodore
Users Club
c/o DJ & Son Electronics
Colonial Plaza
Latrobe, PA 15650
Jim Mathers

Commodore Users Club
3021 Ben Venue Dr.
Greensburg, PA 15601
(412) 836-2224
Jim Mathers

VIC 20 Programmers, Inc.
c/o Watson Woods
115 Old Spring Rd.
Coatesville, PA 19320
Robert Gougher

G.R.C. User Club
300 Whitten Hollow Rd.
New Kensington, PA 15068
Bill Bolt

NADC Commodore Users Club
248 Oakdale Ave.
Horsham, PA 19044
Norman McCrary

CACC (Capitol Area Commodore
Club)
134 College Hill Rd.
Enola, PA 17025
(717) 732-2123
Lewis Buttery
Union Deposit Mall at 7 p.m.

G/C Computer Owners Group
c/o Gilbert Associates, Inc.
P.O. Box 1498
Reading, PA 19607
Extension 6472
Jo Lambert (215) 775-2600

Boeing Employees Personal
Computer Club
The Boeing Vertol Co.
P.O. Box 16858
Philadelphia, PA 19142
(215) 522-2257
Jim McLaughlin

South Central PA Commodore Club
2109 Cedar Run Dr.
Camp Hill, PA 17011
(717) 763-4219
David Persing

Main Line Commodore Users
Group (MLCUG)
c/o Main Line Computer Center
1046 General Allen Lane
West Chester, PA 19380
(215) 388-1581
Emil Volcheck

Commodore Users Group
781 Dick Ave.
Warminster, PA 18974
Matt Matulaitis

The Commodore Users Club
of S.E. Pittsburgh
c/o Groves Appliance & TV
2407 Pennsylvania Ave.
West Mifflin, PA 15122
Charles Groves

Compstars
130 Blue Teal Circle
Audubon, PA 19403
Mike Norm
Meet at Audio Video Junct.

Scranton Commodore Users Group
P.O. Box 211
Clarks Summit, PA 18411

Clifton Heights Users Group
P.O. Box 235
Clifton Heights, PA 19018

Oxford Circle 64 Users Group
Frankford Cong. Un. Church of Christ
Oxford Ave. & Pratt St.
Philadelphia, PA 19124
(215) 743-8999
Roger Nazeley (215) 535-9021
4th Wed. of Month

VIC Software Development Club
440 W. Sedgwick
Apt. A-1
Philadelphia, PA 19119
(215) 844-4328
Tracy Lee Thomas

Bits & Bytes
1015 Dale Rd.
Secane, PA 19018
(215) 544-5875
Dave Boodey

CACCC-Centre Area Commodore
Computer Club
214 Computer Building
University Park, PA 16802
(814) 237-5912
Bill Hillner

PUERTO RICO

CUG of Puerto Rico
RFD #1, Box 13
San Juan, PR 00914
Ken Burch

VIC 20 User Group
655 Hernandez St.
Miramar, PR 00907
Robert Morales, Jr.

RHODE ISLAND

Irving B. Silverman, CPA
160 Taunton Ave.
E. Providence, RI 02914
Contact: Michelle Chavanne

Newport VIC/64 Users
10 Maitland Ct.
Newport, RI 02840
(401) 849-2684
Dr. Matt McConeghy

The VIC 20 Users Club
Warwick, RI 02886
Tom Davey

Commodore Users Group
c/o Data-Co.
978 Tiogue Ave.
Coventry, RI 02816
(401) 828-7385
Victor Moffett

SOUTH CAROLINA

Beaufort Technical College
100 S. Ribaut Rd.
Beaufort, SC 29902
Dean of Instruction

Computer Users Society
of Greenville (CUS)
Horizon Records-Home Computers
347 S. Pleasantburg Dr.
Greenville, SC 29607
(803) 235-7922
Bo Jeanes

Commodore Computer Club
of Columbia
318 Quincannon Dr.
Columbia, SC 29210
Buster White Sect./Treas.

Spartanburg Commodore
Users Group
803 Lucerne Dr.
Spartanburg, SC 29302
(803) 582-5897
James Pasley

SOUTH DAKOTA

PET User Group
515 South Duff
Mitchell, SD 57301
(605) 996-8277
Contact: Jim Dallas

VIC/64 Users Club
608 West 5th
Pierre, SD 57501
(605) 224-4863
Larry Lundeen

TENNESSEE

River City Computer
Hobbyists
Memphis, TN
1st Mon. at Main Library

Nashville Commodore Users Group
P.O. Box 121282
Nashville, TN 37212
(615) 331-5408
Dave Rushing
3rd Thurs. at Cumberland Mus

Commodore User Club
Metro Computer Center
1800 Dayton Blvd.
Chattanooga, TN 37405
Mondays 7:30 pm

Metro-Knoxville Commodore
Users Club
7405 Oxmoor Rd., Rt. #20
Knoxville, TN 37921
(615) 938-3773
Ed Pritchard

Memphis Commodore Users Group
2476 Redvers Ave.
Memphis, TN 38127
(901) 358-5823
Harry Ewart

TEXAS

PET Users
2001 Bryan Tower
Suite 3800
Dallas, TX 75201

Larry Williams
P.OD Box 652
San Antonio, TX 78293

PET User Group
John Bowen
Texas A & M
Microcomputer Club
Texas A & M, TX

CHUG (Commodore Houston
Users Group)
8738 Wildforest
Houston, TX 77088
(713) 999-3650
Contact: John Walker

Commodore Users Group
5326 Cameron Rd.
Austin, TX 78723
(512) 459-1220
Dr. Jerry D. Frazee

VIC Users Group
3817 64th Dr.
Lubbock, TX 79413

64 Users Group
2421 Midnight Circle
Plano, TX 75075
S.G. Grodin

Savid Computer Club
312 West Alabama
Suite 2
Houston, TX 77006
Davi Jordan, Chairman

Gulf Coast Commodore
Users Group
P.O. Box 128
Corpus Christi, TX 78403
(512) 887-4577
Lawrence Hernandez

Mid-Cities Commodore Club
413 Chisolm Trail
Hurst, TX 76053
Bruce Nelson

Interface Computer Club
814 North Sabinas
San Antonio, TX 78207
M. E. Garza, President

ICUG (Irving Commodore
Users Group)
3237 Northgate #1289
Irving, TX 75062
(214) 252-7017
Robert Hayes

Commodore Computer Club (C3)
c/o Lamar Full Gospel Assembly
1200 S. Sumner
Pampa, TX 79065
(806) 665-3444
Randy Mills
Every other Thurs. 7 p.m.

UTAH

Utah PUG
Jack Fleck
2236 Washington Blvd.
Ogden, UT 84401

The Commodore Users
Club
742 Taylor Avenue
Ogden, UT 84404
Contact: Todd Woods Kap,
President
David J. Shreeve,
Vice President

The VIClic
799 Ponderosa Drive
Sandy, UT 84070
Contact: Steve Graham

VIC 20 Users
324 N. 300 W.
Smithfield, UT 84335
Dave DeCorso

Northern Utah VIC & 64
Users Group
P.O. Box 533
Garland, UT 84312
David Sanders

The Commodore Users Group
652 West 700 North
Clearfield, UT 84015
(801) 776-3950
Rodney Keller, Richard Brenchly

Mountain Computer Society
P.O. Box 1154
Sandy, UT 84091
Dave Tigner

VIRGINIA

Northern VA PET Users
Bob Karpen
2045 Eakins Court
Reston, VA 22091
(803) 860-9116

VIC Users Group
Rt. 2, Box 180
Lynchburg, VA 24501
Contact: Dick Rossignol

VIC Users Group
c/o Donnie L. Thompson
1502 Harvard Rd.
Richmond, VA 23226

Dale City Commodore
User Group
P.O. Box 2004
Dale City, VA 22193
(703) 680-2270
James Hogler

Tidewater Commodore
Users Group
4917 Westgrove Rd.
Virginia Beach, VA 23455
Fred Monson

Fredericksburg Area
Computer Enthusiasts
P.O. Box 324
Locust Grove, VA 22508
(703) 972-7195
Michael Parker

VIC 20 Victims
4301 Columbia Pike #410
Arlington, VA 22204
(703) 920-0513
Mike Spengel

Peninsula Commodore 64
Users Group
124 Burnham Place
Newport News, VA 23606
(804) 595-7315
Richard G. Wilmoth

Norfolk Users Group
1030 West 43rd St. B-4
Norfolk, VA 23508
489-8292
Larry Pearson

NASA VIC 20 User Group
713 York Warwick Dr.
Yorktown, VA 23692
Harris Hamilton

135 Beverley Rd.
Danville, VA 24541
David Gray

R.A.C.E. Commodore Users Group
4726 Horseman Dr.
Roanoke, VA 24019
(703) 362-3960
Larry Rackow

Commodore Users of Franklin
1201 N. High St.
Franklin, VA 23851
(804) 562-6823
D. Bruce Powell

WASHINGTON

NW PET Users Group
2565 Dexter N. 3203
Seattle, WA 98109
Contact: Richard Ball

PET Users Group
c/o Kenneth Tong
1800 Taylor Ave. N102
Seattle, WA 98102

Whidbey Island Commodore
Computer Club
947 N. Burroughs Ave.
Oak Harbor, WA 98277
Michael D. Clark

Central Washington
Commodore Users Group
1222 S. 1st St.
Yakima, WA 98902
Tim McElroy

Blue Mountain Commodore
Users Club
15 Stone St.
Walla Walla, WA 99362
(509) 525-5452
Keith Rodue

Spokane Commodore User Group
N. 4311 Whitehouse
Spokane, WA 99205
(509) 328-1464
Stan White

CBM Users Group
803 Euclid Way
Centralia, WA 98531
(206) 736-4085
Rick Beaber

Computer Club
c/o Honeywell, Inc.
5303 Shilshole Ave., NW
Seattle, WA 98107
(206) 789-2000
John Goddard

WEST VIRGINIA

Personal Computer Club
P.O. Box 1301
Charleston, WV 25325
Cam Cravens

TriState Commodore Users
73 Pine Hill Estates
Kenova, WV 25530
(304) 453-2124
Marc Hutton

Commodore User Group
73 Pine Hill Estates
Kenova, WV 25530
(304) 453-2124
Marc Hutton

WISCONSIN

Sewpus
c/o Theodore J. Polozynski
P.O. Box 21851
Milwaukee, WI 53221

Waukesha Area Commodore
User Group (WACUG)
256½ W. Broadway
Waukesha, WI 53186
Contact: Walter Sadler
(414) 547-9391

Commodore 64 Software
Exchange Group
P.O. Box 224
Oregon, WI 53575
E. J. Rosenberg

C.L.U.B. 84
6156 Douglas Ave.
Caledonia, WI 53108
(414) 835-4645 pm
Jack White
2nd Sat every month 10:00 am

VIC-20 & 64 User Group
522 West Bergen Dr.
Milwaukee, WI 53217
(414) 476-8125
Mr. Wachtl

Menomonie Area Commodore
Users Group
510 12th St.
Menomonie, WI 54751
(715) 235-4987
Mike Williams

C.U.S.S.H.
3614 Sovereign Dr.
Racine, WI 53406
(414) 554-0156
Tim Tremmel
3rd Saturday of month

Madison Area Commodore
Users Group
1552 Park St.
Middleton, WI 53562
(608) 831-4852
John Carvin
3rd Thurs. each month

S.W.I.T.C.H.
W156 N8834 Pilgrim Rd.
Menomonee Falls, WI 53051
(414) 255-7044
Len Lutz

Milwaukee Area CBM64
Enthusiasts (M.A.C.E.)
P.O. Box 340
Elm Grove, WI 53122
(414) 259-5991
Kevin Wilde

The Eau Claire CBM64
Users Group
Rt. 5, Box 179A
Eau Claire, WI 54703
(715) 874-5972
John Slavsky, Jr.
2nd Thurs. 7 p.m.

WAVE
P.O. Box 0641
Waukesha, WI 53187

WYOMING

Commodore Users Club
c/o Video Station
670 North 3rd #B
Laramie, WY 82070
(307) 721-5908
Pamela Nash

CANADA

Toronto PET Users Group, Inc.
1912A Avenue Rd., Ste. 1
Toronto, Ontario, Canada
M5M 4A1
(416) 782-8900 or call 416-782-9252
Contact: Chris Bennett

PET Users Club
c/o Mr. Brown
Valley Heights Secondary School
Box 159
Langton, Ont. N0E 1G0

Vancouver PET Users Group
P.O. Box 91164
West Vancouver, British Columbia
Canada V7V 3N6

CCCC (Canadian
Commodore Computer Club)
c/o Strictly Commodore
47 Coachwood Place
Calgary, Alberta, Canada
T3H 1E1
Contact: Roger Olanson

W.P.U.G.
9-300 Enniskillen Ave.
Winnipeg, Manitoba R2V 0H9
Larry Neufeld

VIC-TIMS
2-830 Helena St.
Trail, British Columbia
V1R 3X2
(604) 368-9970
Greg Goss

Arva Hackers
Medway High School
Arva, Ontario N0M 1C0
D. Lerch

Nova Scotia Commodore
Computer Users Group
66 Landrace Cres.
Dartmouth, N.S. B2W 2P9
Andrew Cornwall

Bonnyville VIC Cursors
Box 2100
Bonnyville, Alberta T0A 0L0
(403) 826-3992
Ed Wittchen

Commodore Users Club of Sudbury
938 Brookfield Ave.
Sudbury, Ontario
P3A 4K4

PET Educators Group
P.O. Box 454 Station A
Windsor, Ontario
N9A 6L7

COMVIC
P.O. Box 1688
St. Laurent
Montreal, Quebec
H4L 4Z2

Calgary Commodore Users Group
37 Castleridge Dr., N.E.
Calgary, Alberta
T3J 1P4
John Hazard

Fledging Barrie User Group (BUG)
58 Steel St.
Barrie, Ontario
Canada L4M 2E9

INTRUDER ALARM
for VIC 20TM or Commodore 64TM

BUZZER
MODEL V100
VIN Systems (USA)
P.O. Box 31
Boone, Iowa 50036
1-515-432-2300

Intruder Interface Includes:
1. V100 Alarm Interface
2. Four Magnetic Switches
3. Piercing 95db Siren
4. 100 ft. twisted wire
5. Alarm software
* More accessories available

Total System Only $69.00
plus $2.75 for postage and handling

Specify Computer and Memory
☐ VIC 20TM ☐ Commodore 64TM
☐ Disk ☐ Cassette

Name ____________
Address ____________
City ________ State ________ Zip ________
Phone Number ( ) ____________
MC or VISA # ____________
Expiration Date ____________

Send Check, M.O., VISA/MC to:

VIN Systems (U.S.A.) P.O. Box 31
Boone, Iowa 50036

VIC 20 and Commodore 64 are trademarks of
Commodore Electronics Ltd.

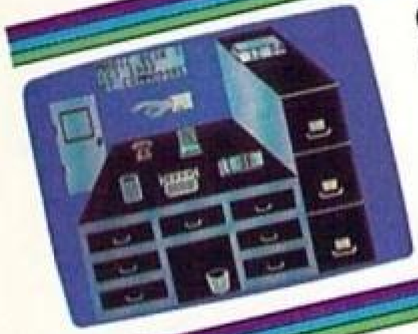
departments

advertisers index

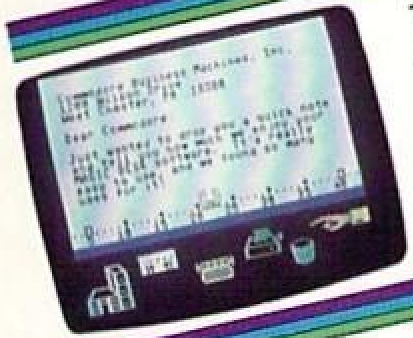
Academy Software	31
Commodore IFC, IBC, BC	6 & 7, 14 & 15
Futurehouse	97
Micro 80	99
Microcomputer Games	9
Orange Micro	33
Peek Magazine	74
Protecto	58-69
Public Domain	99
Sub-Logic Corporation	5 & 13
Vin Systems	128

Commodore 64 Magic Desk I

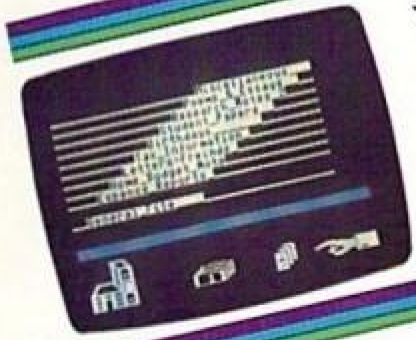
Only From Commodore—The Excitement and Simplicity of Magic Desk!



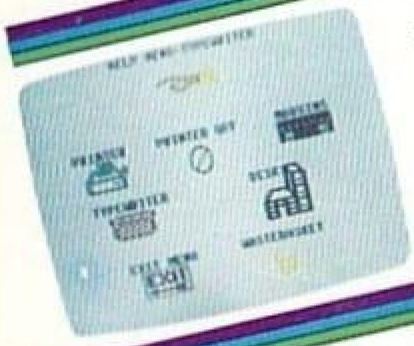
Only Commodore brings you the magic of MAGIC DESK. . . the next generation of "user friendly" software! Imagine using your computer to type, file and edit personal letters and papers *without learning any special commands!* All MAGIC DESK commands are PICTURES. Just move the animated hand to the picture of the feature you want to use (like the TYPEWRITER) and you're ready to go.



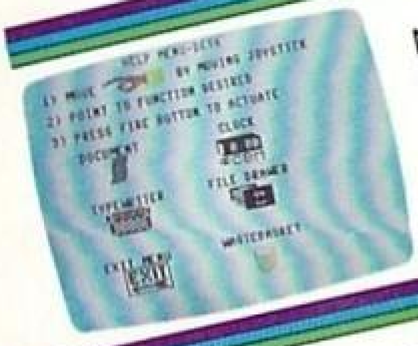
The MAGIC DESK Typewriter works just like a real ELECTRIC TYPEWRITER . . . and it's COMPUTERIZED. All the filing is *electronic*. Excellent sound effects and screen animation make typing fun, whether you're typing letters, reports or memos . . . and the built-in filing feature makes MAGIC DESK useful for keeping names and addresses, home inventory lists, insurance information and more.



Your COMMODORE 64, COMMODORE DISK DRIVE and MAGIC DESK are an unbeatable combination. Filing operations are automatically linked to your Commodore disk drive—but you don't have to know any commands—just "file" the pages you type in the file cabinet and your text is automatically saved on diskette. There are 3 file drawers with 10 file folders in each drawer and 10 pages in each folder.



To PRINT a page you've typed, just "point" at the picture of the printer and your pages are automatically printed on your COMMODORE PRINTER or PRINTER/PLOTTER. If you want to erase what you've typed, the WASTEBASKET under the desk lets you "throw away" pages. There's even a DIGITAL CLOCK which helps you keep track of time while you're typing.


HELP MENU-DESK

1) MOVE BY MOVING JOYSTICK

2) POINT TO FUNCTION DESIRED

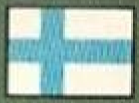
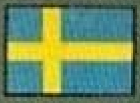
3) PRESS FIRE BUTTON TO ACTUATE

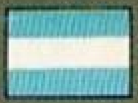
Not only is MAGIC DESK easy to use . . . it's hard to make a mistake! Just press the COMMODORE key and one of several "help menus" appears to tell you exactly what to do next. Special messages show you how the various picture commands work and help you when you make a mistake. Help messages also show you how to use the printer, filing cabinet, digital clock and wastebasket.

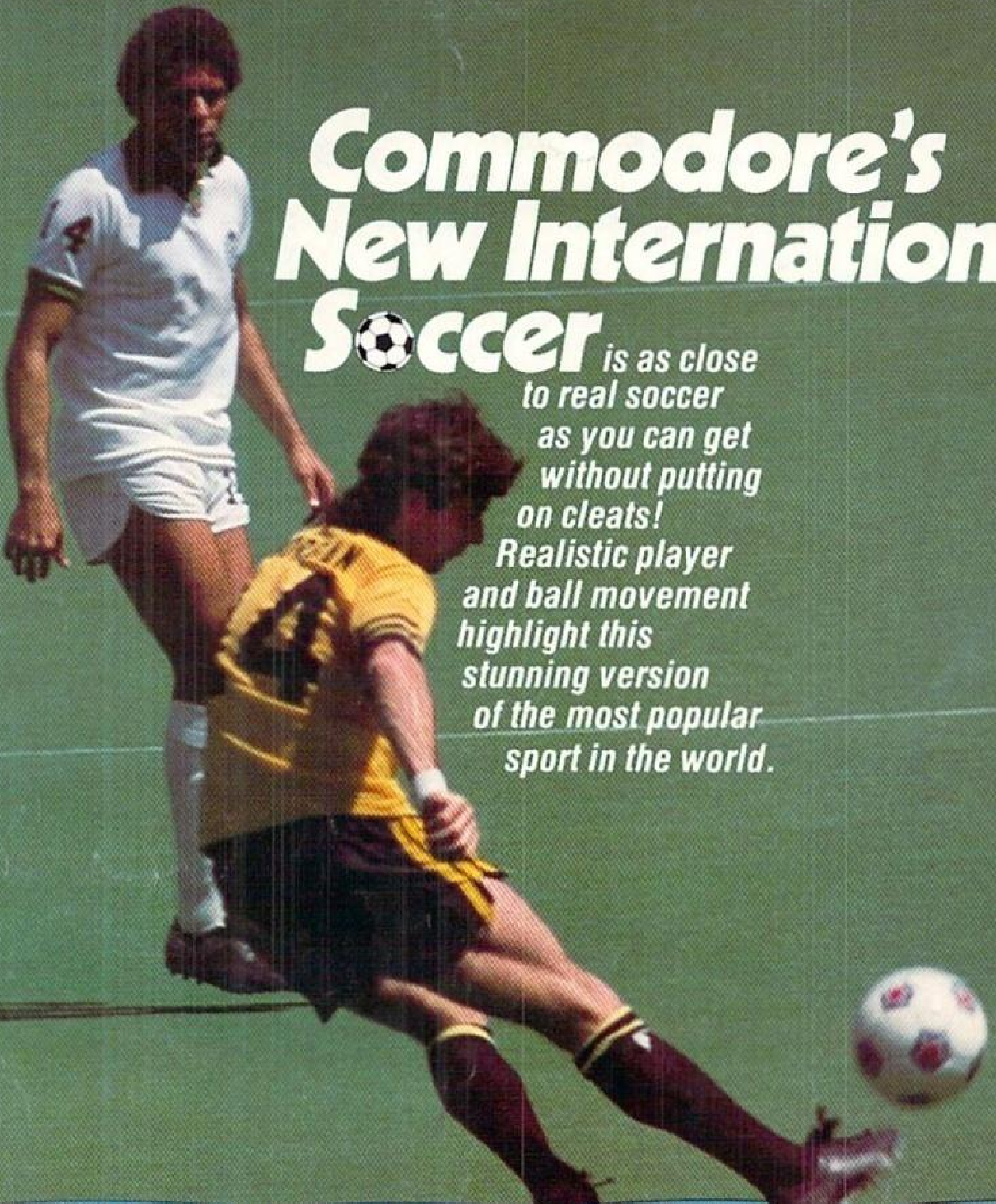
Another reason why Commodore is number one in the world of microcomputers—Come join us.

commodore
COMPUTERS

First In Quality Software

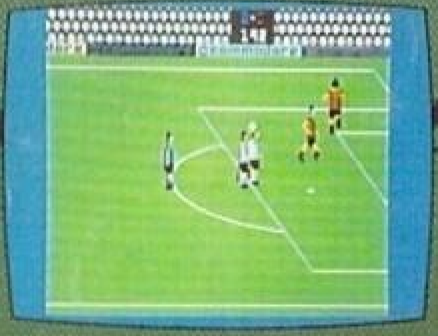





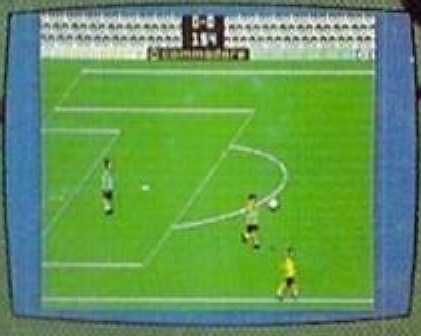


Commodore's New International Soccer

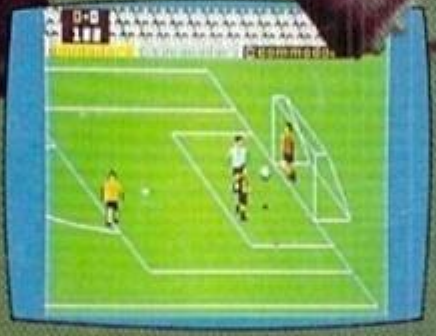
is as close to real soccer as you can get without putting on cleats! Realistic player and ball movement highlight this stunning version of the most popular sport in the world.



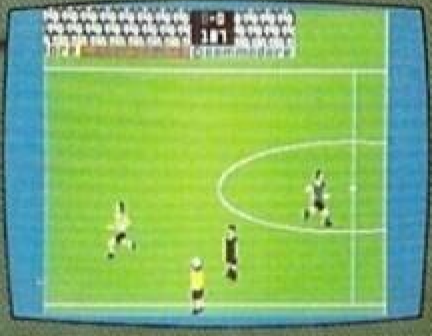
Players can head the ball.



Players chase down a free goal kick.



True-to-life goalkeeping.



Authentic overhead throw-ins.

commodore COMPUTERS

First In Quality Software

ISBN 0-88731-009-6